

**DEFINICIÓN DE UN PROCESO DE DESARROLLO DE SOFTWARE CON
CONTROL DE CALIDAD DEL PRODUCTO EN UNA EMPRESA PYME DE LA
REGIÓN**

PRESENTADO POR

JHON ALEXANDER HOLGUÍN BARRERA

**UNIVERSIDAD AUTÓNOMA DE MANIZALES
FACULTAD DE INGENIERÍA
MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE
MANIZALES
2014**

**DEFINICIÓN DE UN PROCESO DE DESARROLLO DE SOFTWARE CON
CONTROL DE CALIDAD DEL PRODUCTO EN UNA EMPRESA PYME DE LA
REGIÓN**

PRESENTADO POR

JHON ALEXANDER HOLGUIN BARRERA

**Tesis de Grado para optar por el Título de Magister en Gestión y Desarrollo de
Proyectos de Software.**

ASESOR

Ing. Sandra Victoria Hurtado Gil

Magister En Ingeniería de Sistemas y Computación

**UNIVERSIDAD AUTÓNOMA DE MANIZALES
FACULTAD DE INGENIERÍA
MAESTRÍA EN GESTIÓN Y DESARROLLO DE PROYECTOS DE SOFTWARE
MANIZALES
2014**

Dedicatoria

*A Dios que por su misericordia y gracia,
puso en mi vida una familia que me ha dado
tanto amor, apoyo y paciencia
para culminar este trabajo
"los amo con mi corazón"*

Tabla de contenido

INTRODUCCIÓN	1
1. REFERENTE CONTEXTUAL	3
1.1. ÁREA PROBLEMÁTICA	3
1.2. ANTECEDENTES	5
1.2.1. ESTUDIOS DE INTEGRACIÓN DE PSP CON METODOLOGÍAS AGILES	5
1.2.2. PSP y SCRUM	7
1.2.3. ADAPTACIÓN DE RUP PARA PSP	9
1.2.4. SCRUM Y CMMI	10
1.2.5. ANALIZANDO MARCOS SPI (SOFTWARE PROCESS IMPROVEMENT) PARA APOYAR LA CALIDAD DEL PRODUCTO	11
2. JUSTIFICACIÓN	13
3. OBJETIVOS	16
3.1. OBJETIVO GENERAL	16
3.2. OBJETIVOS ESPECÍFICOS	16
4. REFERENTE TEÓRICO	17
4.1. METODOLOGÍAS DE DESARROLLO DE SOFTWARE	17
4.2. MÉTODOS AGILES	17
4.2.1. SCRUM COMO MARCO DE TRABAJO	19
4.3. PSP (Personal Software Process)	21
4.4. SWEBOK	23
4.5. CALIDAD EN EL SOFTWARE	25
4.6. GQM (Goal Question Metric)	28
5.2. RESULTADOS E IMPACTO ESPERADOS	34
6. DESARROLLO	36
6.1. FASE DE DIAGNÓSTICO	36
6.1.1. DESCRIPCIÓN PROCESO ACTUAL DE LA ORGANIZACIÓN	36
6.1.2. HERRAMIENTAS USADAS	37
6.1.3. VALORES DE MEDICIONES ACTUALES	38
6.1.3.1. CANTIDAD DE DEFECTOS REPORTADOS POR VERSIÓN	40
6.1.3.3. ENCUESTA DE PERCEPCIÓN DEL EQUIPO	48
6.1.4. ACTIVIDADES CUMPLIDAS EN SCRUM Y PSP	53
6.1.5. FORTALEZAS Y PUNTOS DE MEJORA DEL PROCESO ACTUAL	54
6.1.6. SELECCIÓN DE MODELO DE CALIDAD	55

6.2.	FASE DE ANÁLISIS, ADAPTACIÓN E INTEGRACIÓN DE PSP, SCRUM Y EL MODELO DE CALIDAD	62
6.2.1.	CARACTERIZACIÓN DE SCRUM Y PSP vs SWEBOK	62
6.2.2.	MAPEO DE SCRUM Y PSP	72
6.2.3.	INTEGRACIÓN DE SCRUM Y PSP	75
6.3.	RESULTADOS FASE PILOTO	82
6.3.1.	HERRAMIENTAS UTILIZADAS	82
6.3.2.	RESULTADOS DEL <i>SPRINT</i>	86
6.3.3.	RESULTADOS DE CALIDAD DEL INCREMENTO DEL SPRINT.....	93
6.4.	CONCLUSIONES.....	101
7.	BIBLIOGRAFÍA.....	103

Lista de Figuras

FIGURA 1. COSTES DE LA CALIDAD DEL SOFTWARE EN UN PROYECTO DE TAMAÑO MEDIO	5
FIGURA 2. EL CICLO DE VIDA DE SCRUM-PSP	8
FIGURA 3. COMPONENTES DE LA ITERACIÓN SCRUM-PSP	8
FIGURA 4. CMMI AND SCRUM GANANCIAS DE PRODUCTIVIDAD	10
FIGURA 5. ANÁLISIS DE ISO 90003 Y CMMI-DEV v1.2 CON LAS CARACTERÍSTICAS DE LA ISO 25010	12
FIGURA 6. AGILE METHODOLOGY USED.....	18
FIGURA 7. RESULTADOS DE ENCUESTA DE ACOGIDA DE SCRUM EN YAHOO!	19
FIGURA 8. RADIOGRAFÍA DE SCRUM.....	20
FIGURA 9. FLUJO DE TRABAJO DE PSP	23
FIGURA 10. EVOLUCIÓN DEL PROCESO PSP FUENTE: PSP(sm): A SELF-IMPROVEMENT PROCESS FOR SOFTWARE ENGINEERS (HUMPHREY, WATTS S., 2005)	23
FIGURA 11. COSTOS DE DETECCIÓN DE ERRORES VS TIEMPO	25
FIGURA 12. MODELO DE CALIDAD ISO/IEC 9126-1.....	26
FIGURA 13. PROCESO DE EVALUACIÓN ISO/IEC 14598-5	28
FIGURA 14. EL PARADIGMA GQM	28
FIGURA 15. DIAGRAMA DE ACTIVIDADES DE SCRUM Y PSP EN LA ORGANIZACIÓN ACTUALMENTE	37
FIGURA 16. GRÁFICO DE REGISTRO DE ERRORES DE ACUERDO A VERSIÓN REPORTADA DE LA APLICACIÓN	41
FIGURA 17. RESULTADOS, ¿HACE CUÁNTO HACE USO DE NUESTROS PRODUCTOS?	42
FIGURA 18. RESULTADOS, SELECCIONE LAS APLICACIONES QUE SU ORGANIZACIÓN MANEJA CON LA ORGANIZACIÓN.....	43
FIGURA 19. RESULTADOS, ¿CUÁLES PROCESOS DE LA APLICACIÓN TIENE IMPLEMENTADOS EN SU ORGANIZACIÓN?	43
FIGURA 20. RESULTADOS, ¿CONSIDERA QUE LAS APLICACIONES DE LA ORGANIZACIÓN CUMPLEN CON LAS NECESIDADES DE SU ORGANIZACIÓN?	44
FIGURA 21. RESULTADOS, EN GENERAL COMO CALIFICA LA CALIDAD DE LAS APLICACIONES QUE SU ORGANIZACIÓN MANEJA CON LA ORGANIZACIÓN.....	44
FIGURA 22. RESULTADOS, ¿CADA CUÁNTO RECIBE ACTUALIZACIONES DE LAS APLICACIONES DE LA ORGANIZACIÓN?.....	45
FIGURA 23. RESULTADOS, CUANDO RECIBE ACTUALIZACIONES DE LAS APLICACIONES DE LA ORGANIZACIÓN, POR LO GENERAL LA ACTUALIZACIÓN.	45
FIGURA 24. RESULTADOS, CUANDO REPORTA UN ERROR DE LAS APLICACIONES DE LA ORGANIZACIÓN, ¿CUÁNTO DEBE ESPERAR PARA QUE SE CORRIJA ESTE ERROR?	46
FIGURA 25. RESULTADOS, CUANDO RECIBE LA CORRECCIÓN A UN ERROR QUE REPORTÓ DE LAS APLICACIONES DE LA ORGANIZACIÓN, POR LO GENERAL:.....	46
FIGURA 26. RESULTADOS, ¿RECOMENDARÍA LAS APLICACIONES DE LA ORGANIZACIÓN A OTRAS EMPRESAS?.....	47
FIGURA 27. RESULTADOS PREGUNTA, ¿CONSIDERA QUE ACTUALMENTE EXISTE UN PROCESO DEFINIDO DE DESARROLLO DE SOFTWARE DENTRO DE LA ORGANIZACIÓN?	48
FIGURA 28. RESULTADOS PREGUNTA, ¿CONSIDERA QUE POSEE LA SUFICIENTE CAPACITACIÓN DE LA APLICACIÓN DEL MARCO DE TRABAJO DE SCRUM DENTRO DE LA ORGANIZACIÓN?.....	49
FIGURA 29. RESULTADOS PREGUNTA, ¿CONSIDERA QUE POSEE LA SUFICIENTE CAPACITACIÓN DE LA APLICACIÓN DE PSP DENTRO DE SU PROCESO DE DESARROLLO DE SOFTWARE?	49
FIGURA 30. RESULTADOS PREGUNTA, ¿CONSIDERA QUE EL MARCO DE TRABAJO SCRUM Y EL MODELO DE PROCESO PSP ESTÁN INTEGRADOS DENTRO DE UN PROCESO DEFINIDO DE DESARROLLO DENTRO DE LA ORGANIZACIÓN?	50
FIGURA 31. RESULTADOS PREGUNTA, ¿CONSIDERA QUE EL PROCESO DE DESARROLLO ESTA GUIADO POR UN MODELO DE CONTROL DE CALIDAD ESTÁNDAR DEFINIDO?	50
FIGURA 32. RESULTADOS PREGUNTA, ¿CONSIDERA QUE EXISTE LA SUFICIENTE DOCUMENTACIÓN PARA LA EJECUCIÓN DE LAS ACTIVIDADES DE DESARROLLO ASIGNADAS?	51
FIGURA 33. RESULTADOS PREGUNTA, CONSIDERA QUE LA ESTIMACIÓN DE LAS TAREAS DE DESARROLLO DENTRO DE LOS PROYECTOS SON.....	51
FIGURA 34. RESULTADOS PREGUNTA, ¿CONSIDERA LA APLICACIÓN DE SCRUM Y PSP DENTRO DEL PROCESO DE DESARROLLO, FAVORECE AL AMBIENTE DE TRABAJO DEL EQUIPO?	52
FIGURA 35. RESULTADOS PREGUNTA, CUÁL ES LA PERCEPCIÓN DE CALIDAD QUE POSEE DEL PRODUCTOS DESARROLLADOS POR LA ORGANIZACIÓN?	52

FIGURA 36. ESTRUCTURA DE LA DIVISIÓN DE MÉTRICAS DE CALIDAD, FUENTE: AN INFORMATION MODEL FOR SOFTWARE QUALITY MEASUREMENT WITH ISO STANDARDS P. 2 (ABRAN, AL QUTAISH, DESHARNAIS, & HABRA, 2005)	57
FIGURA 37. RELACIÓN ENTRE LAS MÉTRICAS DE LOS TIPOS DE CALIDAD.....	58
FIGURA 38. PORCENTAJE DE MAPEO DE PSP Y SCRUM VS SWEBOK, FUENTE: ELABORACIÓN PROPIA	69
FIGURA 38. VISTA GENERAL DE LAS ÁREAS DEL CONOCIMIENTO DEL SWEBOK	70
FIGURA 39. ÁREA DEL CONOCIMIENTO REQUERIMIENTOS DE SOFTWARE	70
FIGURA 40. ÁREA DEL CONOCIMIENTO DISEÑO DE SOFTWARE	71
FIGURA 41. ÁREA DEL CONOCIMIENTO CONSTRUCCIÓN DEL SOFTWARE.....	71
FIGURA 42. ÁREA DEL CONOCIMIENTO PRUEBAS DE SOFTWARE	71
FIGURA 43. ÁREA DEL CONOCIMIENTO GESTIÓN DE LA INGENIERÍA DE SOFTWARE.....	72
FIGURA 44. ÁREA DEL CONOCIMIENTO CALIDAD DEL SOFTWARE	72
FIGURA 45. DIAGRAMA DE PROCESO GENERAL SCRUM-P	75
FIGURA 46. DIAGRAMA DE PROCESO <i>SPRINT PLANNING</i> SCRUM-P FUENTE: ELABORACIÓN PROPIA	77
FIGURA 47. DIAGRAMA DE PROCESO PLANEACIÓN PSP	77
FIGURA 48. DIAGRAMA DE PROCESO DE DESARROLLO DE TAREAS DEL SPRINT	79
FIGURA 49. DIAGRAMA DE PROCESO DESARROLLO DETALLADO DE TAREA	80
FIGURA 50. DIAGRAMA DE PROCESO POSTMORTEM DE TAREA	81
FIGURA 51. PANTALLA HERRAMIENTA JIRA PARA DEFINIR PRODUCT BACKLOG	84
FIGURA 52. PANTALLA DE EJEMPLO HERRAMIENTA FISHEYE	84
FIGURA 53. HERRAMIENTA DE PSP PROPIETARIA, FUENTE: ELABORACIÓN PROPIA	84
FIGURA 54. INTERFAZ DE USUARIO AGREGADA A LA HERRAMIENTA UCC LOC, FUENTE : ELABORACIÓN PROPIA	86
FIGURA 55. GRAFICA DE <i>BURNDOWN</i> CHART DEL SPRINT PARA LA PRUEBA PILOTO	87
FIGURA 56. GRAFICA DE ERROR DE ESTIMACIÓN DE TIEMPO DE EQUIPO DE DESARROLLO	88
FIGURA 57. GRAFICA DE ERROR DE ESTIMACIÓN EN TAMAÑO EQUIPO DE DESARROLLO	89
FIGURA 58. RESULTADOS, CONSIDERA QUE EXISTE UNA MEJOR INTEGRACIÓN DE SCRUM Y PSP CON RESPECTO AL PROCESO MANEJADO ANTERIOR?	90
FIGURA 59. RESULTADOS, CONSIDERA QUE LAS ACTIVIDADES DEFINIDAS EN EL PROCESO DE SCRUM Y PSP INTEGRADO	90
FIGURA 60. RESULTADOS, CONSIDERA QUE EL PROCESO DE SCRUM Y PSP INTEGRADO ESTA	91
FIGURA 61. RESULTADOS, CONSIDERA QUE LA ESTIMACIÓN GUIADA POR PSP CON RESPECTO A LA ACTIVIDAD DEL SPRINT PLANNING EN SCRUM	91
FIGURA 62. RESULTADOS, CONSIDERA QUE EL PROCESO INTEGRADO DE SCRUM Y PSP IMPACTA EL PROCESO DE DESARROLLO..	91
FIGURA 63. RESULTADOS, CONSIDERA QUE PSP INTEGRADO CON SCRUM IMPACTA EL CONTROL DE CALIDAD DEL DESARROLLO	92
FIGURA 64. RESULTADOS, CONSIDERA LA APLICACIÓN DEL PROCESO INTEGRADO DE SCRUM Y PSP DENTRO DEL PROCESO DE DESARROLLO, FAVORECE AL AMBIENTE DE TRABAJO DEL EQUIPO?	92
FIGURA 66. RUTA DE PRUEBAS GENERADO POR LA HERRAMIENTA MCCABE (MCCABE SOFTWARE, 2014)PARA LA FUNCIONALIDAD CON MAYOR COMPLEJIDAD CICLOMÁTICA (47).....	96
FIGURA 67. GRAFICO KIVIAT DE FUNCIONALIDAD DEL SPRINT PRODUCIDO POR MCCABE IQ	97
FIGURA 65. GRAFICA DE RESULTADOS EVALUACIÓN IT-MARK.....	99

Lista de Tablas

TABLA 1. REPORTE DE EVALUACIONES Y VALORACIONES DE CMMI (PARTE I).....	3
TABLA 2. REPORTE DE EVALUACIONES Y VALORACIONES DE CMMI (PARTE II)	4
TABLA 3. COMPONENTES DE LOS COSTOS DE CALIDAD	4
TABLA 4. RESULTADOS DE BÚSQUEDA DE ARTÍCULOS RELACIONADOS CON METODOLOGÍAS ÁGILES Y PSP.....	6
TABLA 5. ANÁLISIS DE LA SUBCARACTERÍSTICA DE CALIDAD EXACTITUD CON ISO 90003 Y CMMI.....	11
TABLA 6. CARACTERÍSTICAS DE ISO 25000 Y ASPECTO QUE ATIENDE CADA UNA.	27
TABLA 7. REGISTRO DE ESTIMACIÓN DE TIEMPO VS TIEMPO REAL POR PROYECTO.....	30
TABLA 8. ESTIMACIÓN DE TIEMPO VS TIEMPO REAL POR SPRINT.....	30
TABLA 9. REGISTRO DE ESTIMACIÓN DE TIEMPO VS TIEMPO REAL POR TAREA	30
TABLA 10. DEFINICIÓN DE ACTIVIDADES CUMPLIDAS EN SCRUM Y PSP.....	30
TABLA 11. ERRORES DETECTADOS EN VERSIONES Y TIEMPO DE CORRECCIÓN.....	30
TABLA 12. EVALUACIÓN DE MODELOS DE CALIDAD.....	31
TABLA 13. EVALUACIÓN DE CARACTERÍSTICAS RELEVANTES DE MODELO DE CALIDAD SELECCIONADO	31
TABLA 14. MAPEO DE SCRUM Y PSP FRENTE A LAS KA Y TÓPICOS DESCRITOS EN EL SWEBOK.....	31
TABLA 15. COMPARACIÓN DE ROLES, ACTIVIDADES Y ARTEFACTOS DE PSP Y SCRUM PARA INTEGRACIÓN Y SELECCIÓN	32
TABLA 16. RESULTADOS ESPERADOS EN EL PROYECTO.....	34
TABLA 17. IMPACTOS ESPERADOS A FUTURO POR EL PROYECTO	35
TABLA 18. ROLES ACTUALES DE LA ORGANIZACIÓN	36
TABLA 19. HERRAMIENTAS UTILIZADAS EN LA ORGANIZACIÓN	38
TABLA 20. REGISTRO DE ESTIMACIÓN DE TIEMPO VS REAL POR PROYECTO FUENTE: ELABORACIÓN PROPIA	38
TABLA 21. REGISTRO DE ESTIMACIÓN DE TIEMPO VS REAL POR SPRINT	39
TABLA 22. REGISTRO DE NÚMERO DE TAREAS PLANEADAS POR SPRINT	39
TABLA 23. RESUMEN DE TABLA DE REGISTRO DE TIEMPOS DE TAREAS, TOMADO DEL ANEXO 1 (REGISTRO DE ESTIMACIÓN DE TIEMPO VS REAL POR TAREA) FUENTE: ELABORACIÓN PROPIA.....	40
TABLA 24. REGISTRO DE ERRORES DE ACUERDO A VERSIÓN REPORTADA DE LA APLICACIÓN.....	41
TABLA 25. FICHA TÉCNICA ENCUESTA "EVALUACIÓN DE CALIDAD DEL PRODUCTO"	42
TABLA 26. FICHA TÉCNICA ENCUESTA "PERCEPCIÓN DEL PROCESO DEL EQUIPO DE DESARROLLO"	48
TABLA 27. ACTIVIDADES DE SCRUM Y PSP, ACTUALMENTE REALIZADAS EN LA ORGANIZACIÓN.....	54
TABLA 28. ESCALA EVALUACIÓN MODELO DE CALIDAD	55
TABLA 29. CALIFICACIÓN CUALITATIVA PARA SELECCIÓN DE MODELO DE CALIDAD.....	56
TABLA 30. DIVISIONES NORMA ISO/IEC 25000.....	57
TABLA 31. MAPEO DE CARACTERÍSTICAS NORMA ISO VS GQM FUENTE: ELABORACIÓN PROPIA.....	59
TABLA 32. SELECCIÓN DE CARACTERÍSTICAS RELEVANTES DE LA NORMA A CUMPLIR	59
TABLA 33. TIPOS DE MÉTRICAS ESTABLECIDAS PARA NORMA DE CALIDAD.....	60
TABLA 34. FOCOS DE MÉTRICAS ESTABLECIDAS PARA NORMA DE CALIDAD	60
TABLA 35. MEDIDAS DE FUNCIONALIDAD SELECCIONADAS NORMA ISO 25023.....	60
TABLA 36. MEDIDAS DE RENDIMIENTO SELECCIONADAS NORMA ISO 25023	61
TABLA 37. MEDIDAS DE USABILIDAD SELECCIONADAS NORMA ISO 25023	61
TABLA 38. MEDIDAS DE MANTENIBILIDAD SELECCIONADAS NORMA ISO 25023.....	62
TABLA 39. SCRUM MAPEADO BAJO LOS LINEAMIENTOS DEL SWEBOK	64
TABLA 40. PSP MAPEADO BAJO LOS LINEAMIENTOS DEL SWEBOK	69
TABLA 41. CARTA DE COLORES DE MAPEO SCRUM Y PSP FRENTE AL SWEBOK.....	70
TABLA 42. TIPO DE RELACIONES DE ACTIVIDADES, ROLES Y ARTEFACTOS DE SCRUM Y PSP	73
TABLA 43. ACTIVIDADES SELECCIONADAS DE SCRUM Y PSP PARA INTEGRACIÓN SCRUM-P	75
TABLA 44. HERRAMIENTAS AJUSTADAS Y AGREGADAS AL PROCESO INTEGRADO DE SCRUM-P	83
TABLA 45. MÉTRICAS DE MEDIDA PSP MODIFICADA, FUENTE: ELABORACIÓN PROPIA.....	85
TABLA 46. CLASIFICACIÓN DE PROXYS PARA ELABORACIÓN DE TABLA RELATIVA DE TAMAÑOS.....	85
TABLA 47. EJEMPLO DE ORGANIZACIÓN DE PROXYS Y CLASIFICACIÓN POR TAMAÑO	86
TABLA 48. NUMERO DE TAREAS PLANEADAS VS COMPLETADAS	86
TABLA 49 FICHA TÉCNICA ENCUESTA DESARROLLADORES POSTERIOR A LA IMPLEMENTACIÓN DE SCRUM-P	90

TABLA 51. RESULTADOS MEDICIONES CRITERIOS DE CALIDAD DE FUNCIONALIDAD.....	93
TABLA 52. RESULTADOS MEDICIONES CRITERIOS DE CALIDAD DE RENDIMIENTO.....	94
TABLA 53. RESULTADOS MEDICIONES CRITERIOS DE CALIDAD DE USABILIDAD.....	94
TABLA 54. RESULTADOS MEDICIONES CRITERIOS DE CALIDAD DE MANTENIBILIDAD	95
TABLA 55. RESUMEN DE RESULTADOS MEDICIONES CRITERIOS DE CALIDAD, FUENTE: ELABORACIÓN PROPIA.....	97
TABLA 50. RESULTADOS DE EVALUACIÓN IT MARK, CMMI NIVEL 2	98

Lista de Anexos

ANEXO 1. TABLA DE REGISTRO DE ESTIMACIÓN DE TIEMPO VS REAL POR TAREA	106
ANEXO 2. RESULTADOS PREGUNTAS ENCUESTA DE EVALUACIÓN DEL PRODUCTO	112
ANEXO 3. RESULTADOS DETALLADOS ENCUESTA DE EQUIPO DE DESARROLLO	116
ANEXO 4. GUIONES MODIFICADOS DE PSP INTEGRADO CON SCRUM	119
ANEXO 5. RESULTADO RESPUESTAS ENCUESTA EQUIPO DE DESARROLLO PROCESO DE SCRUM-P	160
ANEXO 6 MEDIDAS RECOLECTADAS PARA EVALUACIÓN DE CALIDAD EN FUNCIONALIDAD	162
ANEXO 7 MEDIDAS RECOLECTADAS PARA LA EVALUACIÓN DE CALIDAD EN RENDIMIENTO	164
ANEXO 8. MEDIDAS RECOLECTADAS PARA EVALUACIÓN DE CALIDAD EN USABILIDAD	166
ANEXO 9 MEDIDAS RECOLECTADAS PARA EVALUACIÓN DE LA CALIDAD EN MANTENIBILIDAD	168
ANEXO 10 DESCRIPCIÓN DEL PROCESO DE DESARROLLO ANTERIOR	170
ANEXO 11. KAS Y TÓPICOS DEL SWEBOK NO MAPEADOS CON SCRUM Y PSP	171
ANEXO 12. TABLA DE HERRAMIENTAS UTILIZADAS PARA EL PROCESO DE EVALUACIÓN DE CALIDAD.....	173

RESUMEN

Actualmente la industria del software tiene grandes retos para la construcción y mantenimiento del software en términos de calidad, costos y tiempo. Para esto se han generado diversidad de metodologías, procesos, normas y modelos que permiten afrontar estos retos, pero es complejo que la aplicación de solo un proceso logre cumplir con esto, ya que existe gran variedad de organizaciones productoras de software con diferentes propósitos, clientes y mercados. Es por esto que se puede decir que la clave del éxito está en la correcta elección de las metodologías más apropiadas y su correcta integración al contexto organizacional.

Este estudio contiene la integración de varias metodologías y normas para el proceso de desarrollo de software de una PYME (pequeña y mediana empresa), más específicamente se incorporó el marco de trabajo SCRUM catalogado como ágil y altamente aceptado a nivel de la industria, junto con el modelo de mejora de procesos PSP (personal software process) el cual también es reconocido por los resultados logrados en las organizaciones que lo aplican. Además se adicionó al proceso la evaluación del producto, a través de una norma de calidad de producto reconocida a nivel mundial como lo es la ISO 25000. La integración de estos dos modelos y la norma en el proceso pretende mejorar la calidad y productividad de la PYME.

ABSTRACT

Currently, the software industry has big challenges for the construction and maintenance of software in terms of quality, costs and time; for this reason, the industry has generated diversity of methodologies, process, standards and models to address these challenges, but is complex to accomplish this with the single application of only one process that meets the goal. Today there are wide variety of software companies with different purposes, customers and markets, so you can say that the key to meet the goal is choice the correct and most appropriate methodologies and their correct integration into organizational context.

This study contains the integration of several of these methodologies and standards in the software development process of an SME (small an median enterprise), specifically the framework to integrate is SCRUM cataloged like Agile and highly accepted by industry, with the improvement model PSP that is recognized by the results achieved in the organizations that implementing it. Also added to the process the product evaluation through of a product quality standard recognized worldwide such as the ISO 25000. The integration of these two models and the quality standard is aimed to improve the quality and productivity of the SME.

INTRODUCCIÓN

La calidad es una de las características fundamentales en un producto de cualquier índole, y en la industria del software no es la excepción; este atributo hace parte de las exigencias que hoy hace el mercado, como lo define la ISO *"La calidad es la facultad de un conjunto de características inherentes al producto, sistema o proceso para cumplir los requisitos de los clientes y de otras partes interesadas"* (Griful Ponsati & Canela Campos, 2010), de tal manera que se requiere de un proceso organizado que permita satisfacer esta necesidad. Para esto actualmente en la industria del software se puede encontrar numerosa información sobre procesos, conocimientos, disciplinas y marcos de referencia para la mejora de procesos, que permiten llevar a cabo un proceso de construcción de software ordenado con control de la calidad.

Debido a la gran variedad de requerimientos, escenarios, restricciones o limitantes que se pueden presentar a la hora de iniciar un proyecto de desarrollo de software, es difícil concebir que todos usen un mismo proceso de desarrollo, para lo cual existen diversidad de procesos definidos que permitan adaptarse a los requerimientos de cada proyecto (Cusumano, MacCormack, Kemerer, & Crandall, 2003). Igualmente se han formalizado distintas metodologías para el desarrollo de software reconocidas en la industria, y se puede decir que se enmarcan en dos grandes grupos: las "tradicionales" (Priolo, 2009), caracterizadas por su gran rigurosidad en el proceso en la cual se definen una serie de actividades, roles y artefactos, las cuales han resultado ser efectivas en proyectos de gran tamaño y exigencia de cumplimiento de formalidades durante el proceso (Grupo ISSI, 2003); por otro lado están las "ágiles", las cuales buscan minimizar la burocracia en el proceso, con el fin de adaptarse a las necesidades cambiantes de los clientes, sin perder el enfoque de la calidad. (Beedle, y otros, 2001)

A través del desarrollo del presente proyecto, se realizó un estudio de la integración de las actividades definidas en el marco de trabajo SCRUM (Sutherland & Schwaber, 2011) (clasificado como "ágil") al proceso de desarrollo de software de una empresa PYME de la región, que ya tiene aplicación y conocimiento parcial del uso del mismo, con el modelo de proceso de desarrollo de software personal PSP 2.0 (Humphrey, The Personal Software Process (PSP), 2000) establecido por el SEI; esta integración se realizó de acuerdo al análisis de las fortalezas y falencias de SCRUM y PSP, que permitieran establecer cómo las actividades de cada uno se complementan y cuáles son necesarios suprimir por su redundancia o poca utilidad dentro del proceso. De igual forma se analizaron las posibles falencias de documentación existentes, con el fin de presentar alternativas para suplir las mismas, a través de un lenguaje de modelado de software.

Además, como aporte del aseguramiento de la calidad se escogió el estándar de calidad de producto de software ISO 25000, con el fin de verificar la calidad de los productos desarrollados a través del uso de SCRUM y PSP.

El desarrollo del proyecto posee diferentes fases en las que se podrá encontrar inicialmente la caracterización de SCRUM y PSP como marco de trabajo y proceso de desarrollo de software, mapeado con áreas definidas en el cuerpo de conocimientos de la ingeniería de software (SWEBOK, Software Engineering Body of Knowledge) (IEEE, Computer Society, 2004) y la adecuación para su fusión de acuerdo a las necesidades de la organización; por otro lado se tiene el análisis con el cual se seleccionó la norma de calidad ISO para el control de calidad del producto de software, que permitió ser una herramienta para la evaluación y control del producto en cuanto a calidad dentro del proceso de desarrollo de software de la organización. Por último se presenta la definición del proceso propuesto, el cual se procedió a validar con su implementación en un proyecto piloto seleccionado dentro de la organización, que permitió documentar y analizar los resultados de la adaptación de los marcos de trabajo y normas de calidad, con el fin de que sirvan como referencia y caso de estudio para otras organizaciones que posean las mismas características.

El presente documento corresponde al informe final del desarrollo del proyecto, en el cual se presentan los resultados de la integración de SCRUM, PSP y el modelo de calidad ISO, en donde se pudo evidenciar cómo la integración de estos impacta positivamente en la gestión y control del proceso de desarrollo, así como en su calidad, complementándose entre sí a través de sus diversas actividades, especialmente en lo que respecta a la estimación; por otro lado se pudo encontrar que el uso de herramientas de apoyo para la automatización del proceso es un factor determinante que ayuda al cumplimiento de las actividades definidas de toma de métricas.

1. REFERENTE CONTEXTUAL

1.1. ÁREA PROBLEMÁTICA

En Colombia la apertura de mercados a través de tratados de libre comercio ha abierto las posibilidades para que la exportación e importación de productos hacia y desde otros países se facilite (PROEXPORT, 2011), permitiendo que las empresas Colombianas ofrezcan sus productos en mercados externos, y abriendo las puertas para que nuevas competencias del exterior entren en el mercado interno, obligando a aumentar nuestro nivel de competitividad en cuanto a precios, variedad y calidad (PROEXPORT, 2012); La industria del software no es ajena a esta exigencia, y si las empresas locales no están preparadas para enfrentar estos retos, la desaparición de estas en el mercado puede ser una consecuencia.

La calidad es una exigencia que deben cumplir las empresas de software de todo tamaño (Pequeña, Mediana y Grande) para la satisfacción de sus clientes tanto a nivel nacional como internacional, además permite evitar sobrecostos en el proceso de desarrollo de software (Ruiz Carreira & Ramos Roman, 2001). Las empresas en el mundo, conscientes de esto, han empezado a evaluar y mejorar sus procesos de desarrollo de software, esto se evidencia al analizar la siguiente tabla donde se encuentra el número de solicitudes de evaluación y valoraciones emitidas por el SEI (Software Engineering Institute) en una de las más reconocidos modelos de valoración y mejora de procesos en el mundo: CMMI (Capability Maturity Model Integration) (Chrissis, Konrad, & Shrum, 2010):

Country	Appraisals	Maturity Level					Country	Appraisals	Maturity Level				
		1	2	3	4	5			1	2	3	4	5
Angola	10 or fewer						Germany	77	9	32	36	0	0
Argentina	87	0	46	34	0	7	Ghana	10 or fewer					
Australia	14	0	4	8	0	2	Greece	10 or fewer					
Austria	10 or fewer						Guatemala	10 or fewer					
Bahrain	10 or fewer						Hong Kong	16	0	5	7	0	4
Bangladesh	12	0	0	12	0	0	Hungary	10 or fewer					
Belarus	10 or fewer						India	662	1	31	481	5	144
Belgium	14	1	6	7	0	0	Indonesia	10 or fewer					
Brazil	174	1	88	73	1	11	Iraq	10 or fewer					
Brunei Darussalam	10 or fewer						Ireland	10 or fewer					
Bulgaria	10 or fewer						Israel	17	1	0	11	0	5
Canada	45	1	16	22	1	5	Italy	67	0	33	33	0	1
Chile	50	2	24	19	0	5	Jamaica	10 or fewer					
China	2426	1	83	2,208	66	68	Japan	191	8	49	112	9	13
Colombia	48	1	17	20	3	7	Jordan	10 or fewer					
Costa Rica	10 or fewer						Kenya	10 or fewer					
Croatia	10 or fewer						Korea, Republic of	198	0	68	107	14	9
Cuba	10 or fewer						Kuwait	10 or fewer					
Cyprus	10 or fewer						Latvia	10 or fewer					
Czech Republic	10 or fewer						Lebanon	10 or fewer					
Denmark	10 or fewer						Luxembourg	10 or fewer					
Dominican Republic	10 or fewer						Macedonia	10 or fewer					
Egypt	51	1	26	22	0	2	Malawi	10 or fewer					
El Salvador	10 or fewer						Malaysia	62	0	11	48	0	3
Finland	10 or fewer						Mauritius	10 or fewer					
France	140	3	88	48	0	1	Mexico	165	0	82	68	4	11

Tabla 1. Reporte de Evaluaciones y Valoraciones de CMMI (Parte I)

Fuente: Resultados de Scampi v1.2 / v1.3 Clase A para Marzo de 2013, p. 19 (Software Engineering Institute (SEI), 2013)

Country	Appraisals	Maturity Level					Country	Appraisals	Maturity Level				
		1	2	3	4	5			1	2	3	4	5
Morocco	10 or fewer						South Africa	10 or fewer					
Nepal	10 or fewer						Spain	273	1	158	99	3	12
Netherlands	15	1	4	10	0	0	Sri Lanka	18	0	2	15	0	1
New Zealand	10 or fewer						Sweden	10 or fewer					
Norway	10 or fewer						Switzerland	14	0	10	3	0	1
Pakistan	27	1	21	4	0	1	Syrian Arab Republic	10 or fewer					
Panama	10 or fewer						Taiwan	130	1	53	70	5	1
Paraguay	10 or fewer						Thailand	68	0	18	46	0	4
Peru	24	0	11	11	0	2	Tunisia	10 or fewer					
Philippines	20	0	4	10	1	5	Turkey	30	0	0	27	0	3
Poland	10 or fewer						Ukraine	10 or fewer					
Portugal	27	0	11	13	0	3	United Arab Emirates	15	1	4	8	0	2
Qatar	10 or fewer						United Kingdom	86	3	33	42	1	7
Romania	15	1	4	10	0	0	United States	1521	17	600	825	8	71
Russia	11	0	2	5	0	4	Uruguay	10	0	0	7	0	3
Saudi Arabia	18	0	2	11	2	3	Venezuela	10 or fewer					
Singapore	22	0	5	14	2	1	Viet Nam	37	0	0	32	1	4
Slovakia	10 or fewer												

Tabla 2. Reporte de Evaluaciones y Valoraciones de CMMI (Parte II)

Fuente: Resultados de Scampi v1.2 / v1.3 Clase A para Marzo de 2013, p. 20, (Software Engineering Institute (SEI), 2013)

Como se observa en la tabla, las empresas de Latinoamérica ya han empezado a utilizar modelos de mejora de procesos, aunque la diferencia con respecto a otros países es notoria, esto puede deberse a la tendencia a creer que las pequeñas y medianas empresas poseen muchas deficiencias para implementar modelos de mejora de procesos reconocidos (Richardson, 2002) dada su poca capacidad adquisitiva o de complejidad de los procesos, o también por ser modelos con un contacto muy reciente en los países latinoamericanos.

Este proyecto busca influir positivamente en la calidad de los productos creados por una PYME desarrolladora de software de la región, mediante una mejora de los procesos utilizados actualmente para el desarrollo, partiendo de la base de que el uso de un proceso para la construcción de un producto de software es un punto neurálgico que afecta de manera positiva o negativa en la calidad del producto (Humphrey, Managing the Software Process, 1989). Actualmente la PYME se enfrenta a problemas de calidad y de definición de procesos en la organización, que como consecuencia afectan de manera significativa aspectos como la imagen corporativa y los costos de producción, y que pueden clasificarse de la siguiente manera:

El costo de la calidad posee diversos componentes (GTO Partners in Performance, 2007):

Categoría	Definición	Costes Típicos del Software
Errores Internos	Fallos de la calidad, detectados antes de entregar al cliente	Gestión de Errores, corrección y repetición de pruebas.
Errores Externos	Fallos de calidad detectados después de entregar al cliente	Soporte Técnico, investigación de las quejas, notificación de errores por el cliente, Pérdida de Reputación
Valoración	Descubrimiento de la condición del producto	Pruebas y actividades asociadas, auditorías de control de calidad
Prevención	Esfuerzo de asegurar la calidad del producto	Administración de QA, inspecciones y mejoras al proceso, Entrenamiento

Tabla 3. Componentes de los costos de Calidad

Fuente: Revista Colombiana de Computación, p. 76 (Ruiz Carreira & Ramos Roman, 2001)

De lo anteriormente mencionado, se puede resaltar que a nivel general los costos que pueden afectar de manera más negativa a las empresas desarrolladoras de software son los errores externos, lo cual se puede confirmar en la Figura 1. Aquí se refleja que son los más altos, ya que no solo se ven reflejados en los tiempos de soporte, corrección y pruebas, sino también en una mala imagen que se deja ante los clientes. Por esto es importante prestar atención al control y manejo de la calidad con el fin de reducir las posibilidades de fracaso.

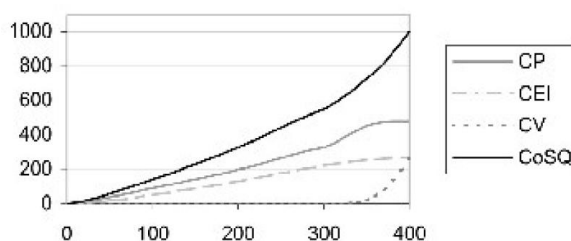


Figura 1. Costes de la Calidad del software en un proyecto de tamaño medio

Fuente: Estimación del coste de la calidad del software a través de la simulación del proceso de Desarrollo, p. 83 (Ruiz Carreira & Ramos Roman, 2001)

- CP: Costos de prevención
- CEI: Costos de errores internos
- CV: Costos de valoración
- CoSQ: Costo de la Calidad

1.2. ANTECEDENTES

Actualmente existen diversos casos de estudio que se han enfocado en la integración de diferentes marcos de trabajo de desarrollo de software, buscando tener como resultado un proceso que aumente las probabilidades de lograr un producto con calidad, a bajo costo y en el tiempo planeado (Yang, Shen, Su, Rong, & Shao, 2011), dentro de estos casos de estudio se ha rescatado algunos en los cuales han utilizado el marco de trabajo SCRUM y el proceso PSP integrado con otros modelos, arrojando resultados satisfactorios, estos casos de estudio pueden servir como referencia de la pertinencia del tema y las posibilidades de éxito que se pueden tener. A continuación se referencian estos casos de estudio y se da una breve descripción de los resultados obtenidos en las mismas.

1.2.1. ESTUDIOS DE INTEGRACIÓN DE PSP CON METODOLOGÍAS AGILES

Actualmente la industria del software posee grandes retos para el desarrollo de sus productos: mejor calidad con bajos costos en el menor tiempo posible, aunque muchos factores pueden influir, existe uno en especial el cual podría decirse que es el más importante "el Proceso" (Yang, Shen, Su, Rong, & Shao, 2011). Hoy en día es posible encontrar diversidad de procesos, pero tal vez es complejo que la aplicación de solo un

proceso logre cumplir con los tres objetivos, es por esto que expertos han empezado a trabajar en la "integración de procesos de software".

Para analizar el nivel de investigación y estudios existentes sobre la integración de procesos de desarrollo de software, se tomará como base un artículo donde se realiza una revisión sistemática al estado del arte sobre este tema, este artículo titulado "Integrando PSP con procesos ágiles: una revisión sistemática" (Yang, Shen, Su, Rong, & Shao, 2011), presenta una serie de conclusiones y resultados de la búsqueda de investigaciones realizadas sobre el tema en diferentes bases de datos digitales y bibliográficas, a continuación se presenta una tabla de resultados de las búsquedas de acuerdo a los criterios (palabras clave) dados en la revisión:

ID	Nombre de Base de Datos	Rpaper	Pre-Resultados	Observacion
1	IEEEExplore	1	7	Búsqueda abstracta
2	ISI Web of Science	2	107	Búsqueda abstracta
3	ProQuest	2	<=80	Abs:(psp) o abs;("personal software process")
4	ScienceDirect	0	40	Abstracta, Título, Palabra clave
5	SpringerLink	2	14	Búsqueda abstracta
6	Kluwer Online	1	35	PSP o "personal software process"
7	CitesserX Library	0	4	Búsqueda abstracta
8	Wiley InterScience	1	27	Búsqueda abstracta
9	Google	3	23	Búsqueda por título
10	ACM Digital Library	4	10	Búsqueda abstracta
11	EBSCOhost	0	15	

Tabla 4. Resultados de búsqueda de artículos relacionados con metodologías ágiles y PSP

Fuente: Integrating PSP with agile process: a systematic review, p. 2 (Yang, Shen, Su, Rong, & Shao, 2011)

Como se puede observar aunque los resultados preliminares (Pre-resultados) de la búsqueda son numerosos, los artículos seleccionados como pertinentes (Rpaper) son muy pocos (En total 12 eliminando los duplicados), por otra parte de la totalidad de los artículos solo uno habla específicamente de SCRUM como metodología ágil, la mayoría del restante trata en general las metodologías ágiles, de esto se puede destacar que aún falta por explorar en este tema y la importancia de su pertinencia. Para el presente proyecto se utilizó uno de los artículos destacado por tratar directamente el tema de integración entre PSP y SCRUM, este artículo lleva por título "*Embracing Process Agility and Discipline*" (Guoping, Dong, & He, 2010) y se trata en una sección posterior.

De este artículo también se resalta cómo las metodologías ágiles proveen un marco de trabajo para la gestión del proceso, mientras que a través de PSP se puede obtener disciplinas para la estimación, planeación y gestión del trabajo individual (Yang, Shen, Su, Rong, & Shao, 2011). Dentro de los beneficios que se logran rescatar en la combinación de metodologías ágiles con PSP se encuentran:

- PSP ofrece mecanismos para las métricas y recolección de datos.
- PSP permite que se realicen estimaciones más razonables.
- PSP ayuda a optimizar los planes.
- PSP ofrece soporte para producir productos de software para cumplir las metas de calidad.
- PSP provee documentación adicional para los métodos ágiles.
- PSP ayuda de manera individual a los ingenieros a mejorar su trabajo personal gradualmente.

1.2.2. PSP y SCRUM

De la conferencia de ingeniería de software APSEC de 2010, se rescata un artículo en el cual se describen los resultados de la integración de SCRUM con PSP (SCRUM-PSP) en un proyecto real. La iniciativa se plantea a raíz de utilizar un modelo disciplinado y dirigido como lo es PSP con el marco de trabajo ágil como SCRUM (Guoping, Dong, & He, 2010).

La implementación del modelo adaptado SCRUM-PSP se realiza en un proyecto en el cual el entorno se adapta de manera favorable a un proceso ágil, ya que existe una disposición constante al cambio de los requerimientos, necesidad de realizar entregas rápidas entre otros, aun así se ve la necesidad de mejorar la estimación y PSP entra como herramienta que permite mejorar este factor, además sirve como herramienta para mejorar el proceso de desarrollo individual de los miembros del equipo.

SCRUM-PSP se diseñó en dos capas; en la primera, "El Ciclo de Vida", en donde se reflejan las iteraciones requeridas para realizar la entrega del producto final (Figura 2), se planea la estrategia de desarrollo, número de iteraciones requeridas, tiempo de cada ciclo y fases en cada iteración (Guoping, Dong, & He, 2010).

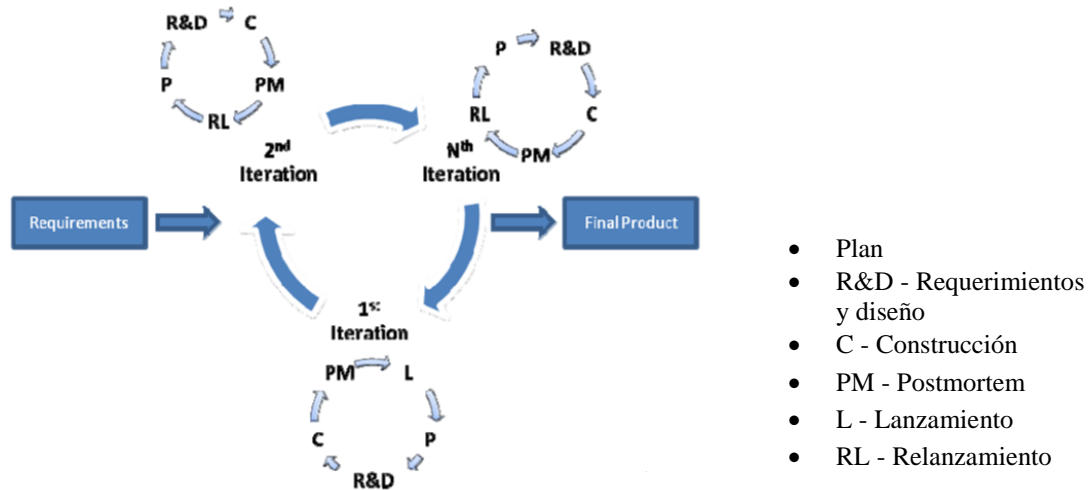


Figura 2. El ciclo de vida de SCRUM-PSP

Fuente: Tomado de SCRUM-PSP: Embracing Process Agility and Discipline, p. 2 (Guoping, Dong, & He, 2010)

La segunda capa llamada iteración SCRUM-PSP, describe los pasos que van dentro de cada iteración (Figura No. 3), son en total 5 pasos seguir:

- Lanzamiento / Re-Lanzamiento
- Plan
- Requerimientos y Diseño
- Construcción
- Postmortem



Figura 3. Componentes de la iteración SCRUM-PSP

Fuente Tomado de SCRUM-PSP: Embracing Process Agility and Discipline, p. 3 (Guoping, Dong, & He, 2010)

Los resultados y conclusiones de este caso de estudio, logran establecer que el marco de trabajo SCRUM puede ser complementado por PSP, permitiendo disponer de las ventajas del uno y el otro. Lo más resaltado de este estudio, según (Yang, Shen, Su, Rong, & Shao, 2011), es:

- Lograr que cada desarrollador tenga la capacidad de predecir o estimar su trabajo, permitirá que el equipo pueda predecir mejor su trabajo.
- La inclusión del plan de calidad en cada iteración, permite que se logren establecer metas de calidad y maximizar la probabilidad de alcanzarlas.
- Permite realizar un seguimiento del proyecto dado por el valor ganado acumulado.
- Las reuniones establecidas en SCRUM (Retrospectiva), permiten al equipo saber el estado del proyecto, posibles desviaciones, estado de la calidad, posibles riesgos.
- SCRUM tiene definidos roles.

A pesar de que existen muy pocos estudios acerca de la integración de SCRUM y PSP, como se mencionó con anterioridad, se puede evidenciar los buenos resultados que se obtiene de la mezcla de dos procesos de software reconocidos.

El presente proyecto no buscó implementar el proceso desarrollado en este estudio citado, ni se utilizó como base para el desarrollo del mismo, sino que se tuvo como referencia. De acuerdo a las necesidades de la organización, se adaptaron las actividades de SCRUM y PSP para integrarse de manera complementaria, permitiendo explotar las ventajas que ofrece cada una; por otro lado uno de los elementos diferenciadores que posee el presente proyecto, es la inclusión de un modelo de control de calidad de producto, para asegurar y evaluar los resultados al aplicar la integración de estos procesos.

1.2.3. ADAPTACIÓN DE RUP PARA PSP

Este es un caso de estudio desarrollado en el Instituto de Computación, Facultad de Ingeniería de la república de Montevideo, Uruguay, en el cual se estudia el uso de un proceso derivado de RUP, llamado MUM (Modelado, Uniforme y Medible) aplicado junto al modelo PSP (Perez Queiruga & Vallespir, 2009), este caso de estudio que tuvo la oportunidad de realizar la adaptación de este modelo de procesos a una extensión de PSP, permitió establecer ciertas ventajas que se evidenciaron en el caso de estudio:

- Inclusión de plantillas para la identificación de tareas y armado del cronograma.
- Listas de chequeo basados en datos personales sobre defectos.
- Inclusión de plantillas para registrar ideas de mejora de proceso.
- Métricas de soporte en la mejora del proceso personal.

Este caso de estudio sirvió como referente para establecer que al igual que los métodos ágiles, también en los métodos tradicionales, PSP es un modelo que permite complementar las metodologías de tal manera que permita que se mejore el proceso de desarrollo de software.

1.2.4. SCRUM Y CMMI

Las metodologías ágiles han presentado resultados exitosos al fusionarse con otros modelos como en este caso lo es CMMI,. En una publicación del manifiesto ágil de 2001, Mark Paul uno de los principales contribuyentes y editor del modelo CMM versión 1.0, observó que el uso de metodologías ágiles, permitía maximizar los beneficios de una buena práctica. (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012)

Uno de los casos de estudio más importantes y que ofrece datos contundentes sobre el uso de CMMI junto con SCRUM, son los resultados obtenidos por la gran empresa proveedora de software Systematic, quienes con productos complejos para los sectores de la salud, defensa nacional y manufactura, hacen una mezcla contundente entre CMMI y SCRUM la cual llaman la "Poción Mágica" (Sutherland, Ruseng Jakobsen, & Johnson, Scrum and CMMI Level 5: The Magic Potion for Code Warriors, 2005), en la cual logran establecer que el re-trabajo de sus proyectos se redujo en un 38% a 42% con un nivel de madurez CMMI nivel 5 y más de 92% de entregas a tiempo a cliente. A continuación se presenta una gráfica de cómo el uso de un modelo de madurez como CMMI Nivel 5 junto con el marco de trabajo SCRUM, ha incrementado la productividad de la organización.

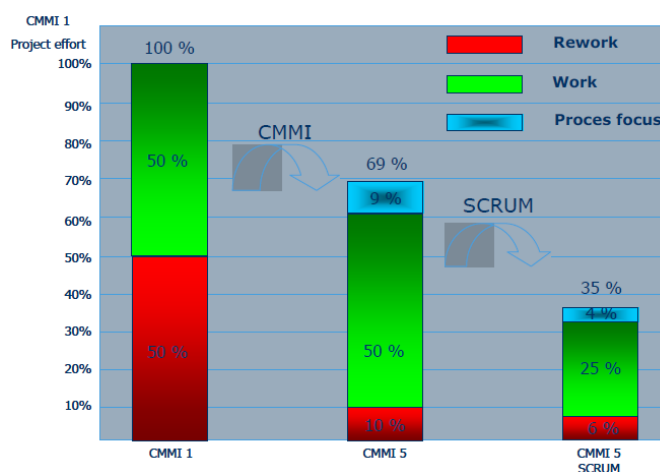


Figura 4. CMMI and Scrum Ganancias de Productividad

Fuente: ScrumPapers, p. 109 (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012)

1.2.5. ANALIZANDO MARCOS SPI (SOFTWARE PROCESS IMPROVEMENT) PARA APOYAR LA CALIDAD DEL PRODUCTO

En las organizaciones la calidad del producto es un factor relevante que les dará una ventaja competitiva frente a su competencia, pero cada organización dependiendo de su área de desempeño puede variar sus requerimientos de calidad de producto a cumplir; este es un estudio que buscó identificar y establecer cómo los elementos de mejora de procesos presentados en los marcos ISO 90003 y CMMI-DEV v1.2, afectan positivamente a la calidad del producto, basándose en las características de calidad establecidas en la norma ISO 25010.

En el artículo se menciona cómo se realizó el estudio a través del mapeo de las subcaracterísticas de la ISO 25010, y como los SPI (mejora de procesos de software) analizados logran satisfacer estas necesidades a través de los elementos que se presentan en cada uno, un ejemplo de esto se puede evidenciar en la siguiente tabla donde se relaciona la exactitud frente a los marcos de mejora de procesos:

Subcaracterística	Cláusulas ISO 90003	Áreas de proceso claves CMMI-DEV	Comentarios
<i>Exactitud. El grado por el cual el producto software provee el correcto o resultado especificado y con el grado de precisión necesario.</i>	7.3.4 Revisión del diseño y desarrollo ... 7.3.5 Verificación del diseño y desarrollo ... 7.3.6 Validación del diseño y desarrollo ... 7.3.6.1 Validación y Pruebas. 7.3.6.2	Objetivos específicos de las áreas de proceso: Integración del Producto (3) Validación (2) Verificación (3)	En ISO 9003 se analizaron las cláusulas que relacionan recomendaciones para verificar o probar el producto software en el cumplimiento del resultado especificado. En CMMI-DEV la integración de los diferentes componentes del producto y su respectiva verificación y validación, conducen al correcto resultado especificado del producto software.

Tabla 5. Análisis de la subcaracterística de calidad exactitud con ISO 90003 y CMMI
Fuente: Marcos SPI como apoyo a las características de calidad del producto, p. 9 (Pardo, Pino, Garcia, & Piattini, 2009)

A través del anterior análisis y la aplicación de otros métodos de calificación se logró obtener la siguiente gráfica donde se puede evidenciar cómo los marcos de mejora de procesos aportan a las características de calidad definidas en la ISO 25010. De esta gráfica se puede resaltar que ISO 90003 es una norma que aporta en mayoría al cumplimiento de estas características frente a CMMI; aun así las dos son herramientas útiles que pueden aportar a que la calidad del producto sea la deseada.

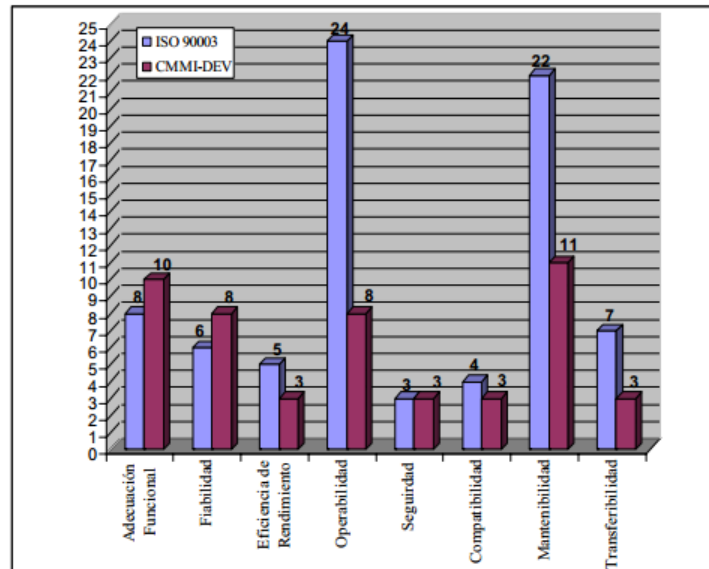


Figura 5. Análisis de ISO 90003 y CMMI-DEV v1.2 con las características de la ISO 25010
Fuente: Marcos SPI como apoyo a las características de calidad del producto, p. 9 (Pardo, Pino, García, & Piattini, 2009)

Este antecedente donde se analizan los marcos SPI para apoyar la calidad del producto permite una vez más afirmar que la aplicación de marcos de trabajo en los procesos de desarrollo de software, impactan de manera positiva a la calidad del producto. PSP provee prácticas y el entrenamiento que son fundamento para otros modelos como TSP (Team software process) (McHale, Chick, & Miluk, 2010) los cuales pueden aportar de manera positiva a las organizaciones a alcanzar el un nivel de madurez CMMI, por ende a la calidad del producto.

Además de los antecedentes presentados, también se tuvieron en cuenta algunos artículos referentes al tema que sirvieron como base y referencia para el desarrollo del presente proyecto, a continuación se relacionan estos artículos sin entrar en detalles:

- PEM: The small company-dedicated software process quality evaluation method combining CMMI and ISO/IEC 14598 (Trudel S. , Lavoie, Paré, & Suryn, 2006)
- SPI Models:What Characteristics are Required for Small Software Development Companies? (Richardson, 2002)

2. JUSTIFICACIÓN

La construcción de software es una de las industrias que está tomando un papel protagónico hoy en la sociedad, la economía y la educación, entre otros sectores; Existe una dependencia cada vez mayor que hace que interactuemos con uno o más sistemas informáticos día a día, minuto a minuto: el sistema operativo de un celular, un buscador en la web, el horno microondas de nuestro hogar, el tablero de control de nuestro automóvil, cada vez más dispositivos controlados por software; este fenómeno no es ajeno a nuestro país, ya que cada vez es menor la brecha digital que permite que accedamos fácilmente a estos dispositivos, también es importante resaltar que la industria de software en nuestro país es un potencial para convertirse en un sector económico importante en el mercado nacional e internacional (PROEXPORT, 2011) (Fedesoft, MINTIC, Vive Digital FiTi, 2012).

Lo anterior hace que surja una pregunta muy importante y lógica en este contexto, *¿Que nos garantiza que estos productos de software cumplan sus tareas correctamente?*, la respuesta tal vez puede ser hallada en otra respuesta a una pregunta hecha directamente a los responsables de la construcción de estos productos de software, *¿Qué proceso se está utilizando para la construcción de estos productos?*; Watts Humphrey hace una reflexión muy importante acerca de este tema y nos confirma correspondencia de estas preguntas, él afirma: *"La calidad de un sistema de software es gobernada por la calidad del proceso usado para desarrollar y mantener el mismo"* (Humphrey, Managing the Software Process, 1989). Un proceso organizado y efectivo para la construcción de software, no solamente se podrá ver reflejado en la calidad del producto final, si no que a su vez lograra solventar problemas muy comunes en los proyectos de software, como sobrecostos en el presupuesto y fallo en las entregas al cliente lo cual será de mucho beneficio para la empresa en sí.

La calidad es uno de los atributos de software que viene directamente ligado a los procesos de construcción del mismo como ya se mencionó anteriormente, y es este atributo que hace que exista un factor competitivo favorable a las empresas que cumplan con las expectativas de los clientes sobre el mismo.

Se seleccionó una PYME desarrolladora de software de la región, en donde el autor del presente proyecto labora, y la cual requería una mejora del proceso de desarrollo de sus productos de software y la implementación de un modelo de control de calidad, que le permitiera solucionar sus problemas de calidad del producto y además tener un proceso de desarrollo definido adecuado a sus necesidades, de tal manera que se alinee para lograr cumplir con el objetivo de calidad dado.

La ingeniería de software permite establecer un conjunto de métodos herramientas y procedimientos orientados a la construcción de software eficiente, que satisfaga las

necesidades del cliente (Cortés Morales, 1998). Así como se afirma, estos procesos o métodos de desarrollo de software, aportan a la mejora de la calidad del producto que disfrutará el cliente, y a optimizar los costos y tiempos de producción del mismo. Actualmente existen diferentes metodologías, métodos y procesos de desarrollo de software como RUP (Rational Unified Process), Scrum, XP (Extreme Programming), AUP (Agile Unified Process), por otro lado modelos de mejora de procesos de software como PSP, TSP, CMMI, IT Mark, y estándares de control de calidad como ISO/IEC 9126, ISO/IEC 12207, SQuaRE ISO/IEC 25000, los cuales cada uno de manera individual pueden dar resultados efectivos en los proyectos de software donde se aplican, no obstante una adecuación e integración de estos procesos puede generar un resultado más efectivo si se aplican en conjunto; El objetivo de este proyecto fue realizar un trabajo de integración entre el marco de trabajo de desarrollo de software SCRUM, el modelo de mejora de procesos de desarrollo PSP y el estándar de control de calidad de producto ISO 25000, adaptado a una empresa PYME desarrolladora de software del Eje Cafetero, con el fin de poder realizar un seguimiento del antes y el después de la aplicación de estos modelos, en un proyecto piloto, y establecer conclusiones que sirvan como caso de estudio útil para otras empresas de la misma categoría.

Se eligió SCRUM como marco de trabajo, debido a su gran flexibilidad y rigurosidad para la gestión del proceso de desarrollo de software, también es el marco que ha sido acogido por la empresa PYME de desarrollo de software durante hace dos años, lo cual hace que no exista una curva de aprendizaje que pueda demorar el proceso, por otro lado, grandes y pequeñas empresas de la industria, como Yahoo, Google, y Microsoft entre otras han acogido esta metodología con excelentes resultados (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012).

PSP se adoptó como un modelo de mejora de procesos de desarrollo de software, por dos razones fundamentales: como primera razón, es un modelo, junto a TSP, desarrollado por el SEI (Software Engineering Institute), enfocados a los procesos de desarrollo de software a nivel individual y de equipo. El SEI se ha propuesto alinear estos modelos con CMMI a través del método AIM (Agile Improvement Method), para lograr valoraciones de CMMI en un menor tiempo y con mayor efectividad. Junto a otros modelos como SIX SIGMA, PSP y TSP dan un amplio cubrimiento a las áreas de procesos propuestas en CMMI (McHale, Chick, & Miluk, 2010). Por otro lado PSP, como su nombre lo dice, está enfocado a mejorar el proceso de desarrollo de software a nivel personal, lo cual es muy importante antes de atacar otras falencias que pueda tener el equipo de desarrollo, primero es necesario garantizar que a nivel individual se está trabajando de una manera correcta y con calidad, que permita establecer que el resto de acciones tomadas sean efectivas para mejorar la calidad de los productos de software.

Como valor diferenciador del presente proyecto frente a otras propuestas que buscan integrar metodologías tradicionales y ágiles, se tiene la inclusión de un modelo de control de calidad ISO dentro de los mencionados anteriormente, que permita validar el éxito de la integración de SCRUM junto a PSP reflejados en la calidad del producto final.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Adaptar e integrar el marco de trabajo SCRUM y el proceso de mejora de procesos PSP, aplicando un estándar de control de calidad de producto de software, en una empresa PYME desarrolladora de software de la región.

3.2. OBJETIVOS ESPECÍFICOS

- Diagnosticar el estado actual de los procesos de desarrollo de software que utiliza la empresa, a través de una descripción cualitativa del proceso actual y el análisis cuantitativo de los registros históricos de desarrollo y nivel de satisfacción de cliente.
- Establecer el estándar de calidad de producto más adecuado a aplicar dentro del proceso, a través del análisis de los modelos ISO/IEC 9126, ISO/IEC 25010 y ISO/IEC 25023, y establecer los lineamientos del mismo.
- Caracterizar la metodología de desarrollo de software SCRUM y el modelo de mejora de proceso personal PSP, a partir de los lineamientos establecidos en el SWEBOK (IEEE, Computer Society, 2004).
- Adaptar e Integrar el marco de trabajo SCRUM y el modelo de mejora de procesos PSP, a través de la selección y supresión de las actividades de cada uno que permitan complementarse.
- Realizar una prueba piloto de la definición del modelo de desarrollo de software adaptado, en un proyecto de software de la PYME, realizando la medición de datos de la prueba a través de GQM, y evaluando los resultados obtenidos.

4. REFERENTE TEÓRICO

4.1. METODOLOGÍAS DE DESARROLLO DE SOFTWARE

A menudo es fácil encontrar problemas en los proyectos de desarrollo de software, sobrecostos en el presupuesto, falta de entregas a tiempo de los resultados esperados y no cubrimiento de las expectativas por parte del usuario final; esto por lo general es la consecuencia de no utilizar o utilizar incorrectamente una metodología o proceso para construir un producto de software, es por esto que es demasiado importante tener en cuenta qué metodologías de se utiliza para la creación de un producto de software.

"La aplicación de un sistemático, disciplinado y cuantificable enfoque a el desarrollo, operación y mantenimiento del software se le conoce como aplicación de la Ingeniería del Software" (IEEE, 1990)

Una metodología de desarrollo de software es un marco de trabajo que describe el proceso de la creación de un producto de software a través de una planeación, estructuración y control del proceso. Por lo general un modelo de procesos de software debe desempeñar 4 roles: (Booch, 1995):

1. Proveer una guía para organizar las actividades del equipo.
2. Especificar qué artefactos deben ser desarrollados y en qué momento.
3. Determinar las tareas de los desarrolladores y el equipo como un todo.
4. Ofrecer criterios para monitorear y medir el producto del proyecto y sus actividades.

Las anteriores características que se encuentran dentro de las metodologías se condensan en roles, actividades y resultados tangibles.

4.2. MÉTODOS AGILES

Los métodos ágiles están tomando un papel protagónico dentro del desarrollo de software, esto se puede deducir de la amplia acogida que están tomando en grandes empresas desarrolladoras de software a nivel internacional tales como Yahoo, Google, Microsoft, SAP, Blizzard entre otras (Albaladejo, 2012) y nacionales tales como PSL (PSL, 2013), por esto sería un error excluirlas dentro de la caracterización de las metodologías que hoy en día están siendo utilizadas para la fabricación de software. Indirectamente influenciadas por principios de las mejores prácticas de la industria japonesa y basados en los principios de desarrollo implementados en empresas como Toyota y Honda a través del conocimiento de Hirotaka Takeuchi e Ikujiro Nonaka (Poppendieck, 2005) quienes han

trabajado en la creación de nuevas técnicas para la entrega más rápida y flexible de productos al comercio, de esta manera se ha logrado que estas metodologías ágiles se impregnen de métodos más prácticos, utilizados para otro tipo de industrias como la automovilística, fabricación de computadoras e impresoras, dando buenos resultados en la fabricación de software. (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012)

Existen diversidad de métodos ágiles que han surgido hasta el momento tales como XP, Cristal Clear, Agile Unified Process (AUP), SCRUM y otras más, pero es esta última en la cual se va a tratar en las próximas paginas por ser una de las metodologías más empleadas en el desarrollo de software (Version ONE, 2013), dentro de empresas pequeñas como grandes (Google, Microsoft, Yahoo y otras más), además de resultados presentados por aquellos que las han incorporado dentro de sus proyectos (SirsiDynix and Starsoft y Google Adwords nombrando los más conocidos) (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012). A continuación se muestran dos gráficas donde se ve puede observar a SCRUM como la metodología más acogida en la industria del software y como los equipos de desarrollo de software de una empresa reconocida (Yahoo!), prefieren SCRUM.

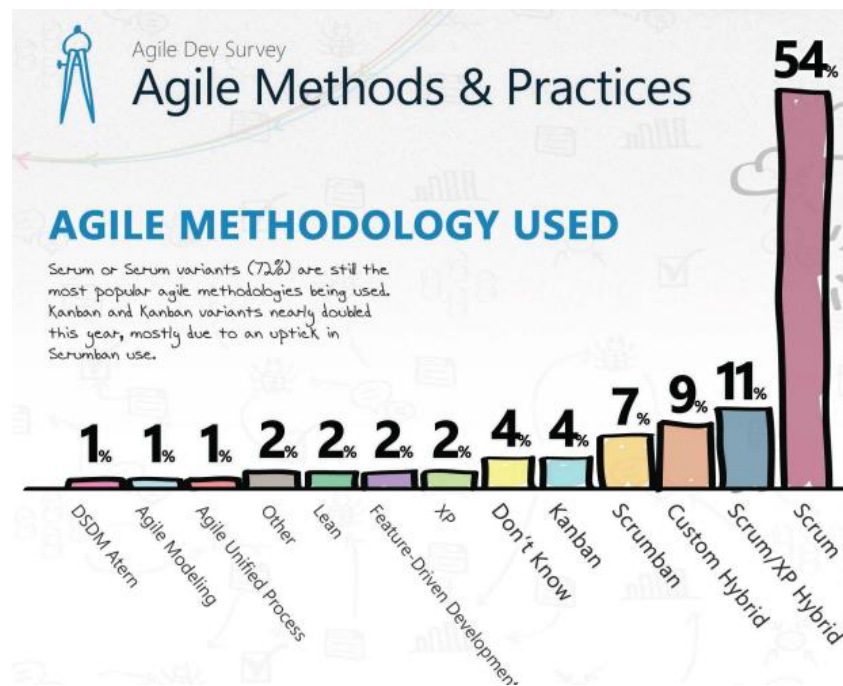


Figura 6. Agile Methodology Used
Fuente: State of Agile Survey 2010, p. 5 (Version ONE, 2013)

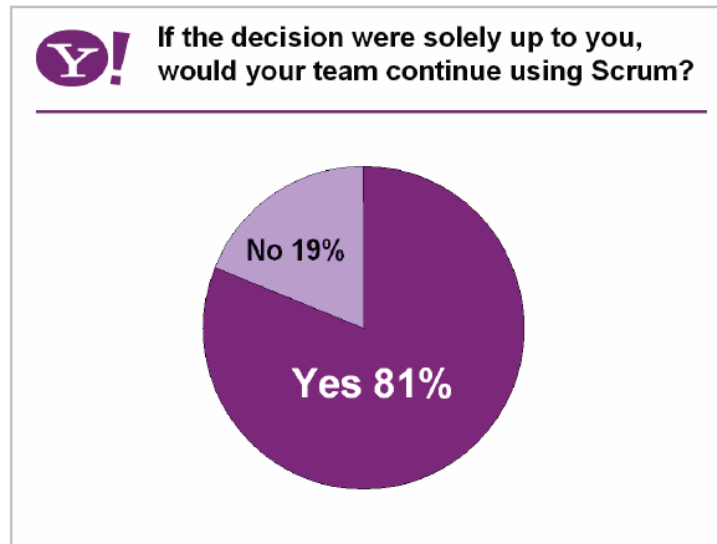


Figura 7. Resultados de Encuesta de acogida de Scrum en Yahoo!
Fuente: Scrum Papers, p. 37 (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012)

SCRUM es un marco de trabajo que busca ofrecer a los clientes y usuarios finales, entregas de software funcional de manera rápida y con calidad (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012); es atractivo por su simplicidad, el proceso en sí, la práctica, los artefactos y reglas que se generan en si son pocos y fáciles de aprender (Schwaber, 2004), adaptándose a proyectos con poca o alta complejidad. Por otro lado se debe entender que este marco no describe lo que se debe hacer en cada momento o situación que se presente dentro del proyecto, esta logra hacer que se pueda tener un panorama total y visible para todos, para así tener la capacidad de tomar decisiones y realizar ajustes en el momento adecuado. A continuación se presenta una descripción del marco de trabajo de SCRUM, sus generalidades, artefactos, procesos y demás para su uso en un proyecto de desarrollo de software.

4.2.1. SCRUM COMO MARCO DE TRABAJO

Es un marco de trabajo a través del cual se busca gestionar el desarrollo de productos complejos de software, donde de una manera creativa y productiva se entrega al cliente o usuario final productos que le representen un alto valor. Este no se define como un proceso o técnica específica para implementar, más bien permite emplear varios procesos y técnicas en el proceso de desarrollo de un producto.

Tiene sus orígenes en los 90's gracias a Jeff Sutherland y Ken Schwaber quienes motivados por las continuas fallas que se presentaban en los proyectos de software que se manejaban a través de una metodología cascada, inician a indagar sobre los procesos desarrollo productivos implementados en otras industrias, especialmente la automotriz y es

allí donde empiezan a referenciar implementaciones hechas por Hirotaka Takeuchi e Ikujiro Nonaka en la industria automotor Japonesa y técnicas implementadas en XP (Jeffries, Anderson, & Hendrickson, 2001) para el desarrollo del marco de trabajo que hoy es conocido como SCRUM (Sutherland J. , The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework, 2012).

Principios

SCRUM está fundamentado en la teoría de control de procesos o también llamado empirismo, el cual determina que el conocimiento proviene de la experiencia y de la toma de decisiones de acuerdo a lo que se sabe. Para que esto funcione es necesario tener claro que esta teoría no funciona correctamente si no se tienen en cuenta tres pilares fundamentales (Sutherland & Schwaber, 2011):

- Transparencia
- Inspección
- Adaptación

Descripción

SCRUM es un marco de trabajo claramente definido, el cual se compone de una serie de artefactos, roles, eventos y reglas en donde cada uno de estos cumple un propósito específico que permitan lograr el objetivo del proyecto (Sutherland & Schwaber, 2011).

A continuación se presenta una gráfica que contiene una descripción resumida de todo el proceso, se encuentran los roles, actividades y artefactos de los cuales se compone esta metodología.

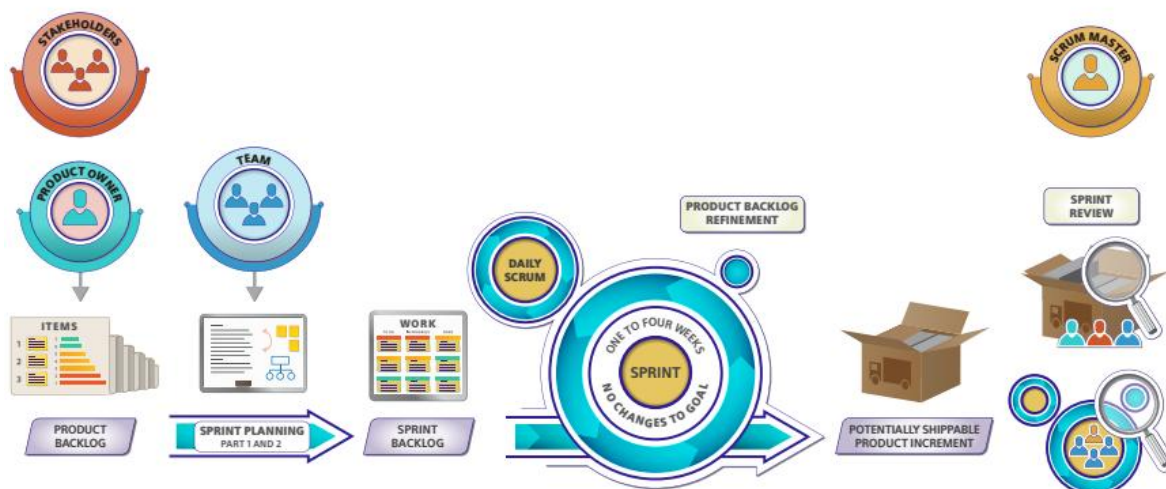


Figura 8. Radiografía de Scrum

Fuente:The Scrum Primer 2.0, p. 3 (Deemer, Benefield, Larman, & Vodde, 2012)

Desglosando la metodología en sus partes principales, tenemos los siguientes puntos.
(Sutherland & Schwaber, 2011)

I. Roles:

- A. Dueño del Producto (ProductOwner)
- B. Equipo (TeamMembers)
- C. Scrum Master

II. Artefactos

- A. ProductBacklog
- B. Sprint Backlog
- C. Burndown Chart

III. Eventos

- A. Sprint Planning
- B. ScrumDaily
- C. Sprint Review

4.3. PSP (Personal Software Process)

PSP (Personal Software Process) es un proceso que como su nombre lo dice permite de manera individual al desarrollador controlar y gestionar la manera en que trabaja, y de esta manera permitirle mejorar la calidad de su trabajo, para esto PSP ha sido diseñado basado una serie de principios de calidad y planeación con el fin de lograr este propósito (Humphrey, The Personal Software Process (PSP), 2000):

- Se debe iniciar de la propuesta de que cada ingeniero es diferente y por lo tanto este es el más indicado para realizar la planeación de su propio trabajo basado en sus propios datos.
- La calidad de los productos depende de la calidad del esfuerzo realizado para producirlos, por ende una de las variables más importantes que influye la calidad de un producto es quien lo construye, es por esto que es importante que los ingenieros encargados de construir el producto estén comprometidos con la calidad del producto y que exista su disposición para luchar por lograr este objetivo.
- Es mucho mejor tratar de prevenir defectos que tratar de buscar una solución para corregirlos.
- Es mucho menos costoso encontrar y corregir errores de manera temprana que tardía.

- Para que los ingenieros puedan mejorar su rendimiento deben manejar un método de medición personal que le permita medirse a sí mismo.
- La forma correcta es siempre la más rápida y barata de hacer las cosas.

Para que los ingenieros puedan realizar su trabajo de la manera correcta es importante que antes de terminar o iniciar se realice un plan a seguir y que este sea desarrollado mediante un método previamente definido. De esta misma manera es importante que los ingenieros puedan establecer cuál es el punto de rendimiento en el cual se encuentran actualmente, a través de la medición de su propio trabajo en cada uno de los pasos que realizan para cumplir una tarea; dentro de esta medición ellos deben establecer los tiempos que gastan en cada paso, el tiempo que gastan en la inyección y corrección de errores, cuáles son los errores más comunes que comenten al realizar su trabajo y realizar un seguimiento a la calidad del producto que están construyendo, una vez todos estos datos sean recolectados cada uno debe analizar sus datos y detectar las fallas que está cometiendo en el proceso para así corregirlas y mejorar a futuro.

Estructura de PSP

PSP está basado en una serie de lineamientos escritos que guían el proceso de principio a fin (Figura 9). Los requerimientos son el estamento de entrada para iniciar el proceso a través de la planeación, que guiará la creación del resumen del plan y la manera de guardar los datos en el mismo. Una vez se tenga el plan, el ingeniero procede a seguir el desarrollo del trabajo en donde registra sus tiempos y defectos inyectados, una vez se finaliza el trabajo se procede a la etapa de *postmortem* donde se resume toda la información obtenida de documentos de registro de tiempos, defectos y tamaño del programa en el resumen del plan creado inicialmente.

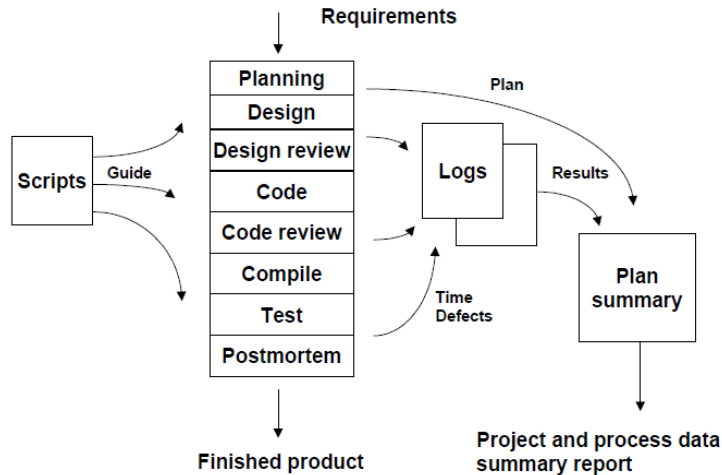


Figura 9. Flujo de Trabajo de PSP

Fuente: The Personal Software Process Report, p. 7 (Humphrey, The Personal Software Process (PSP), 2000)

PSP es un proceso iterativo que es introducido en 6 niveles que van desde PSP0 a PSP2.1, en donde cada una de estos niveles tiene similitud en los artefactos que son manejados en ellas, pero diferencia las temáticas a incorporar en cada nivel, siendo un proceso incremental; dentro de los niveles 0 y 0.1 se introduce la disciplina del proceso y las mediciones, los niveles 1 y 1.1 se introduce la estimación y la planeación, finalmente en el nivel 2 y 2.1 introduce conceptos de calidad (Humphrey, Watts S., 2005); a continuación se presenta una figura que resalta las diferencias un poco más detalladamente:

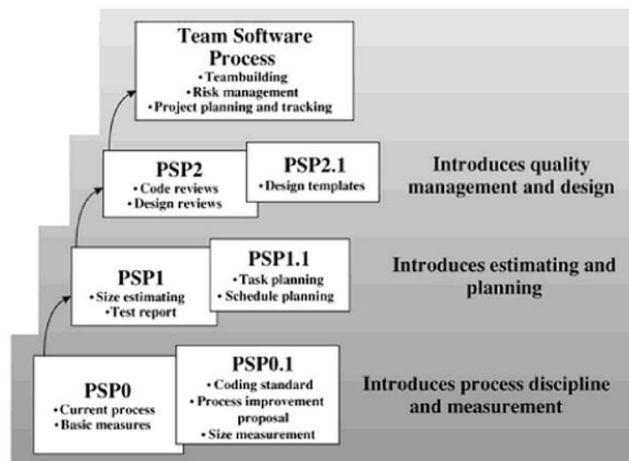


Figura 10. Evolución del proceso PSP Fuente: PSP(sm): A Self-Improvement Process for Software Engineers (Humphrey, Watts S., 2005)

4.4. SWEBOK

SWEBOK (Software Engineering Body of Knowledge) es una cuerpo de conocimientos la cual ha sido desarrollada por la SWECC (Software Engineering Coordinating Committee) y la IEEE Computer Society, en la cual se recopila una porción

organizada de la base del conocimiento de la ingeniería del software, producto de la colaboración de muchas personas, con experiencia en el área de la ingeniería del software. Tiene como propósito lograr cumplir cinco objetivos (IEEE, Computer Society, 2004):

- Promover a nivel mundial una vista consistente de la Ingeniería del Software.
- Clarificar el lugar y Fronteras de la Ingeniería del Software frente a otras disciplinas tales como las ciencias de la computación, gestión de proyectos, ingeniería de computación y matemáticas.
- Caracterizar los contenidos de la Ingeniería de Software como disciplina.
- Proveer un acceso actualizado a la base del conocimiento de la Ingeniería del Software.
- Proveer una base para el desarrollo curricular de la Ingeniería del Software y Material para certificar.

La guía base del conocimiento de la ingeniería del Software (SWEBOK versión 2004), define diez áreas en las cuales se trata de dar una delimitación al enfoque que debe tener la ingeniería del software, fronteras definidas en el objetivo número dos. Estas áreas se les conoce como Áreas del Conocimiento (KA: Knowledge Areas).

- Requerimientos del Software
- Diseño del Software
- Construcción del Software
- Pruebas de Software
- Mantenimiento del Software
- Gestión de Configuración
- Gestión de la Ingeniería del Software
- Gestión de los proceso de Ingeniería
- Gestión de Herramientas y métodos de Ingeniería
- Calidad del Software

Cada una de las áreas del conocimiento se descompone en una serie sub áreas que componen el núcleo de cada área, estas sub-áreas a su vez se descomponen en tópicos y sub-tópicos que la componen.

Para este proyecto las áreas del conocimiento que se consideraron más prioritarias en la construcción de un *proceso de software*¹ definido: requerimientos, diseño, construcción, pruebas, calidad y gestión de la ingeniería del software.

¹ Conjunto de actividades, métodos, practicas y transformaciones que las personas usan para desarrollar y mantener software y sus productos asociados (IEEE, Computer Society, 2004).

4.5. CALIDAD EN EL SOFTWARE

La calidad del software es uno de los componentes que hoy en día está jugando un papel fundamental en la industria y es un factor determinante en la satisfacción del cliente, Phil Crosby la define como "la conformidad de los requerimientos del usuario" (Crosby, 1996) (IEEE, Computer Society, 2004), Watts Humphrey lo define como "Lograr excelentes niveles de aptitud de uso" (IEEE, Computer Society, 2004); es por esto que es necesario adaptar mecanismos y modelos que permitan lograr desarrollar productos de calidad. Por otro lado, sacar un producto de baja calidad genera sobrecostos; se ha logrado establecer que la detección de errores es más costosa proporcionalmente de acuerdo al tiempo transcurrido desde su inyección hasta su detección, esto se puede apreciar en la siguiente gráfica:

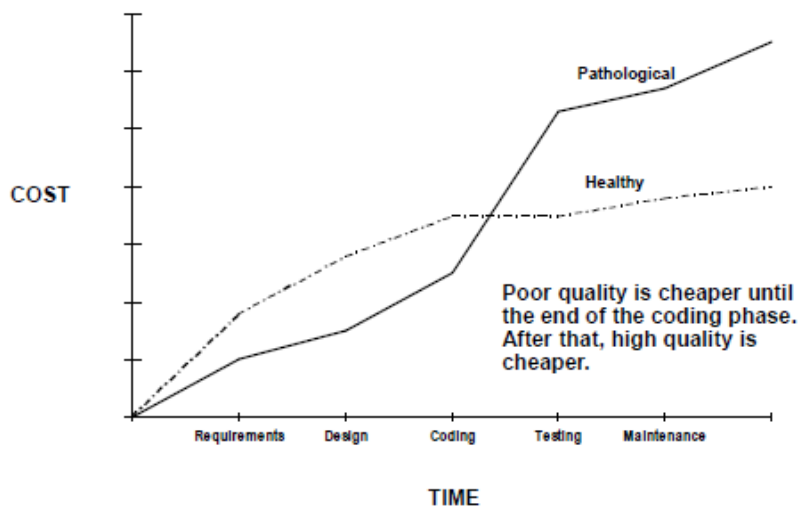


Figura 11. Costos de Detección de Errores vs Tiempo

Fuente: SOFTWARE QUALITY IN 2010, A SURVEY OF THE STATE OF THE ART, p. 15 (Jones, 2010)

De acuerdo a lo anterior se hace evidente que un producto de baja calidad no solo hace que aumente la insatisfacción de los clientes, si no que a su vez es más costoso. La ISO ha definido diversos modelos y estándares de calidad de producto los cuales se han ido adaptando a través del tiempo, para que permitan cumplir con las necesidades de la industria del software, algunos de estos estándares y modelos son:

- ISO/IEC 9126: Es un modelo publicado por primera vez en 1991 y revisado en 2001, se divide en dos partes para evaluar la calidad del producto de software, la primera parte evalúa la calidad interna y externa, la segunda la calidad en uso. Para la calidad interna y externa se especifican seis características que a su vez se subdividen en sus sub-características como se observa en la Figura 12, estos proveen una

listas de aspectos relacionados con la calidad de software (Calero, Moraga, & Piattini, 2010); Actualmente esta norma está siendo incorporada en una nueva serie ISO/IEC 25000.

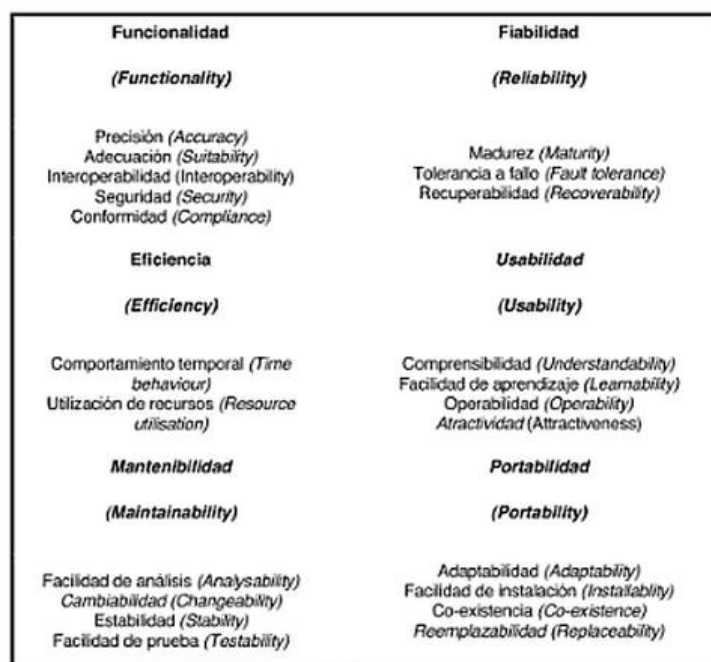


Figura 12. Modelo de calidad ISO/IEC 9126-1

Fuente: (Calero, Moraga, & Piattini, 2010)

- ISO/IEC 25000: SQuaRE (Software Product Quality Requirements and Evaluation) Define una serie de características y sub-características que proveen una terminología para la especificación, medida y evaluación de la calidad de un producto de software, también ISO 25000 en combinación con otros estándares puede ser usada como un marco de trabajo para soportar diferentes estándares (ISO, 2011), esta norma evalúa las características enumeradas a continuación:

Característica	Descripción
Funcionalidad	Grado en que el producto proporciona las funciones que satisfacen las necesidades implícitas y explícitas cuando el producto se utiliza bajo determinadas condiciones.
Confiabilidad	El grado en que un sistema o componente realiza las funciones requeridas bajo condiciones específicas durante un determinado tiempo
Usabilidad	Grado en que el producto puede ser utilizado por usuarios específicos para alcanzar unos objetivos especificados con efectividad, eficiencia y satisfacción en un contexto de uso dado.
Rendimiento/Eficiencia	Rendimiento relativo a la cantidad de recursos utilizados bajo determinadas

	condiciones.
Mantenibilidad	Grado en que el producto puede ser modificado,
Portabilidad	Facilidad con que un sistema o componente puede ser transferido de entorno.
Compatibilidad	Capacidad de dos o más sistemas o componentes para intercambiar información y/o realizar sus funciones requeridas mientras comparten el mismo entorno de hardware o software
Seguridad	Grado de protección de la información y de los datos que hace que personas o sistemas no autorizados no puedan leerlos o modificarlos y que las personas o sistemas autorizados no se les niegue el acceso a los mismos

Tabla 6. Características de ISO 25000 y aspecto que atiende cada una.
Fuente: Calidad de producto y procesos de software (Calero, Moraga, & Piattini, 2010)

- ISO/IEC 14598: Establece un marco que permite evaluar los productos de software a través de la definición de métricas y requisitos para la evaluación de los mismos. Este estándar se divide en seis partes y provee una guía para tres diferentes perspectivas, el desarrollador, el adquiriente y el evaluador, en el estándar. ISO/IEC 14598-5 la parte número 5, describe el proceso de evaluación y las actividades requeridas para realizar una evaluación independiente del producto de software en términos de las características definidas en la ISO/IEC 9126.

La evaluación definida en la norma establece las siguientes actividades:

La figura 13 describe un resumen de la norma ISO/IEC 14598-5

- Análisis de los requerimientos de la evaluación
- Especificación de la evaluación
- Diseño de la evaluación
- Ejecución de la evaluación
- Conclusión de la evaluación

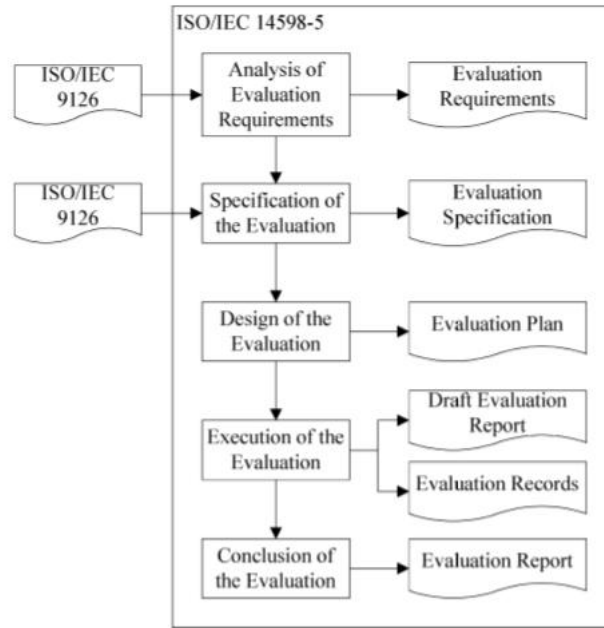


Figura 13. Proceso de evaluación ISO/IEC 14598-5
Fuente p. 5 (Trudel S. , Lavoie, Paré, & Suryn, 2006)

4.6. GQM (Goal Question Metric)

Originalmente es un método creado por V. Basili y D. Weiss, GQM (Meta, Pregunta y Métrica) (Van Soligen & Berghout, 1999), es una metodología de medición enfocada para que la organización establezca primero que todo las metas a cumplir y así definir las métricas efectivas para medir el cumplimiento de las metas, es decir que es orientado a la meta. Una vez se tienen definidas las metas, se procede a refinar estas a través de preguntas a las cuales posteriormente se deberán definir métricas que permitan dar respuesta a estas preguntas planteadas anteriormente. Esta metodología tiene un enfoque *Top-Down* (De lo General a lo Detallado) para la definición, pero la interpretación tiene un enfoque *Bottom-Up* (Del Detalle a lo general) para el análisis de los datos como se muestra en la siguiente figura:

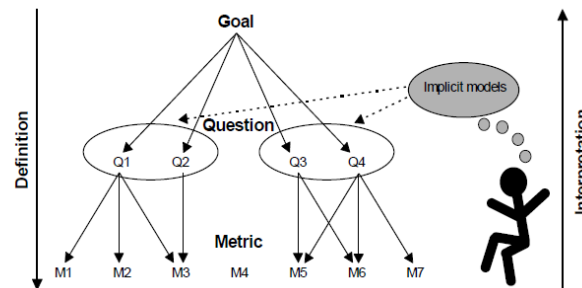


Figura 14. El paradigma GQM
Fuente: The Goal/Question/MetricMethod, p. 23 (Van Soligen & Berghout, 1999)

Este es el modelo a utilizar para definir las mediciones e interpretación de las mismas en el proyecto, debido a su simplicidad y efectividad en los resultados.

5. ESTRATEGIA METODOLÓGICA

5.1. METODOLOGÍA

Se desarrolló una investigación aplicada, relacionada con el proceso de desarrollo de software y la calidad del producto de software en una PYME, con componentes cuantitativos que incluyeron el análisis de datos históricos del proceso de desarrollo de software llevado hasta antes de la aplicación del nuevo proceso, junto a los datos que se recolectaron en la prueba piloto aplicando la integración de los modelos mencionados anteriormente; también hay un componente cualitativo que corresponde a la descripción del proceso anterior y las evaluaciones realizadas a los modelos seleccionados, usando matrices de mapeo y comparación.

El desarrollo del presente proyecto, se efectuó en 5 Fases, de la siguiente manera:

(1) Fase de diagnóstico

Esta fase corresponde al estudio inicial de la organización, su estado actual en cuanto a procesos, metodologías y modelos de calidad utilizados para el desarrollo de software, realizando un análisis descriptivo de cómo se lleva a cabo el proceso de desarrollo de software; para lo cual se usaron los siguientes instrumentos:

- Descripción textual del proceso actual
- Diagrama de Actividades del proceso de desarrollo
- Tabla de roles actuales del equipo de desarrollo
- Cuadro de Herramientas actualmente utilizadas en el proceso de desarrollo.

Se procedió a analizar los datos históricos de los proyectos llevados actualmente en la organización en cuanto a estimaciones, calidad, estado de satisfacción de sus clientes y percepción del equipo de desarrollo. Para esto se utilizaron tablas comparativas de métricas de estos proyectos, recolectadas previamente y el desarrollo de encuestas:

- Encuesta de nivel de satisfacción del producto para clientes
- Encuesta de nivel percepción del proceso a equipo de desarrollo
- Tablas de métricas recolectadas previamente:

Proyecto	Tiempo Estimado (Hrs)	Tiempo Real (Hrs)	Diferencia	Total Tareas Estimadas	Total Tareas con Registro Tiempo	Total Tareas Sin Registro Tiempo

Tabla 7. Registro de estimación de tiempo vs tiempo real por proyecto

Sprint	Tiempo Estimado (Hrs)	Tiempo Real (Hrs)	Diferencia	Total Tareas Estimadas	Total Tareas con Registro Tiempo	Total Tareas Sin Registro Tiempo

Tabla 8. Estimación de tiempo vs tiempo real por Sprint

ID Tarea	Tiempo Estimado (Hrs)	Tiempo Real (Hrs)	Diferencia

Tabla 9. Registro de estimación de tiempo vs tiempo real por tarea

- Tabla de definición y cumplimiento de actividades procesos actuales

SCRUM/ Actividades PSP		
Actividad	Cumple (si/no/parcial)	Observaciones

Tabla 10. Definición de actividades cumplidas en SCRUM y PSP

- Tablas de comparación de número de errores detectados en versiones liberadas y tiempos asignados para corrección.

Versión	No Errores Detectados Soporte	No Errores Detectados Desarrollo	Tiempo asignado Soporte	Tiempo asignado Desarrollo

Tabla 11. Errores detectados en versiones y tiempo de corrección

En esta fase se seleccionó el modelo de calidad de producto que se utilizó en el resto del proyecto. Para la selección del modelo de calidad, se definió un esquema de evaluación que permitió seleccionar el modelo más adecuado para la organización dado por la tabla presentada a continuación. El modelo seleccionado se utilizó para determinar la calidad del

producto de software actual y se evaluó de acuerdo a las características más relevantes entre las normas, las cuales sean comunes entre ellas.

Característica Global	Característica	Norma/ Estándar 1	Norma/ Estándar 2	Norma/ Estándar 3

Tabla 12. Evaluación de modelos de calidad

(2) Fase de análisis, adaptación e integración de PSP, SCRUM y el modelo de calidad

Se obtuvo una radiografía inicial sobre el estado de la organización, sus productos desarrollados y las necesidades de los mismos frente a sus clientes, se procedió a realizar un análisis donde se confrontaron estas necesidades y las actividades definidas en los modelos seleccionados en la integración, esto se dividió en cuatro sub-fases:

- Análisis de las características tratadas en el modelo de calidad seleccionado, y extracción de los puntos álgidos a cumplir según las necesidades de la organización, esta evaluación se realizó a través de una calificación de 0 a 5 de las características, donde 0 significa que no es pertinente y 5 que es totalmente pertinente, de esta manera se permitió determinar cuáles son las más pertinentes, a juicio del investigador y basado en los resultados de la caracterización de la PYME en la fase anterior.

CATEGORÍA	PERTINENCIA (0-5)	SELECCIONADA	OBSERVACIONES

Tabla 13. Evaluación de características relevantes de modelo de calidad seleccionado

- Se realizó un mapeo de las diferentes actividades, roles y artefactos manejados por SCRUM y PSP frente a los lineamientos del SWEBOK de acuerdo a sus áreas del conocimiento (KA) y tópicos, de tal manera que permitió establecer un referente teórico de cada ítem evaluado en los modelos seleccionados.

ROL/ACTIVIDAD/ARTEFACTO	KA	TÓPICO	SUB-TÓPICO	OBSERVACION

Tabla 14. Mapeo de SCRUM y PSP frente a las KA y tópicos descritos en el SWEBOK

GENERALIDAD SWEBOK	ACTIVIDAD SCRUM	ACTIVIDAD PSP	TIPO RELACIÓN	ACTIVIDAD SELECCIONADA	OBS

--	--	--	--	--	--

Tabla 15. Comparación de roles, actividades y artefactos de PSP y SCRUM para integración y selección

- De acuerdo a los resultados obtenidos de las áreas fuertes y débiles de PSP y SCRUM basado en el SWEBOK, se realizó la integración y adaptación de los dos procesos propuestos en un nuevo proceso llamado SCRUM-P (SCRUM-PSP-Modelo de calidad). Para describir el nuevo proceso usado:
 - ✓ Descripción textual
 - ✓ Diagrama de procesos
- Una vez se definió el proceso SCRUM-P se realizó la implementación en los procesos de la organización a través de las siguientes etapas:
 - ✓ Capacitación del equipo de Desarrollo (5 Integrantes)
 - ✓ Definición de Roles dentro del Equipo
 - ✓ Adaptación de las herramientas actuales (de ser necesario)

(3) Fase de prueba piloto

Dividida en tres sub-fases donde su principal objetivo es validar el proceso de implementación del nuevo modelo de SCRUM-P y lanzar su implementación en la organización:

- Selección del proyecto de software de la organización que sirvió como piloto para la implementación de SCRUM-P y el modelo de control de calidad.
- Establecer los mecanismos de medición de acuerdo al modelo de GQM, siguiendo los pasos:
- Establecer Metas, y los objetivos de medidas basados en los objetivos del negocio previamente establecidos teniendo en cuenta:
 - Objeto
 - Propósito
 - Enfoque de Calidad
 - Punto de Vista
- Generación de Preguntas
- Establecer Medidas

- Preparar la recolección de Datos
- Lanzamiento del proyecto guiado por el proceso SCRUM-P e inicio de recolección de datos

(4) Fase de análisis de resultados

Una vez se obtuvieron los datos recolectados sobre la aplicación del proceso definido, se procedió a realizar el proceso de análisis de los resultados obtenidos sobre la implementación, en cuanto a dificultades encontradas en el proceso, experiencia de quienes participaron en el mismo e información sobre el cumplimiento de las expectativas del proyecto, esta recolección de información se realizó a través de entrevistas y grupos focales de manera cualitativa.

(5) Fase de Informe Final y Artículo

El presente proyecto buscó ser un caso de estudio que permita analizar los resultados de la adaptación de varios modelos de desarrollo de software y control de calidad en una empresa de software, además establecer las ventajas, dificultades, resultados que se pueden obtener de la implementación de los mismos, esto con el fin de poder transmitir esto a otras empresas que se puedan encontrar en la misma situación de búsqueda de mejora en sus procesos internos de desarrollo de software, de esta manera se podrán tener en cuenta diversos aspectos que pueden lograr un procesos exitoso en la implementación.

De los resultados obtenidos en la fase 4 se escribió un artículo el cual se envió a una revista indexada con el fin de que esta información sea confrontada con otros expertos del área que puedan realizar aportes y críticas al proyecto.

5.2. RESULTADOS E IMPACTO ESPERADOS

Los resultados esperados una vez finalizado el caso de estudio son los siguientes:

Aspecto	Resultado/Producto	Indicador	Beneficiario
Fortalecimiento de la comunidad académica y la empresa PYME seleccionada.	Documentación sobre la adaptación e integración del marco de trabajo SCRUM, el modelo de mejora de procesos de software PSP, y un modelo de control de calidad del producto de software.	Documento y Diagramas de Procesos (BPN)	Comunidad interesada en el estudio de la ingeniería del software
	Tabla comparativa entre la adaptación de los modelos de desarrollo de software y control de la calidad en la empresa, versus el cubrimiento de las áreas del conocimiento expuestas en el SWEBOK.	Tabla comparativa	
	Documento de Recomendaciones y lecciones aprendidas en el estudio, que sirvan como caso de estudio para otras organizaciones de software PYME de la región.	Documento escrito	
Apropiación social del conocimiento	Artículo científico enviado a una revista indexada sobre el caso de estudio motivo del proyecto.	Documento escrito y constancia de envío	

Tabla 16. Resultados esperados en el proyecto

El desarrollo del presente proyecto permitió ser una guía de caso de estudio para otras empresas desarrolladoras de software de la región, que deseen implementar procesos formales de desarrollo de software y modelos de calidad con el fin incrementar la calidad en sus productos.

A través del estudio de los modelos de desarrollo y calidad de software, junto a su adaptación e implementación en una empresa PYME de la región, se obtuvieron datos importantes sobre las ventajas, desventajas y dificultades de implementación que permitirán

a otras empresas utilizar esta información para mejorar sus procesos de desarrollo y control de la calidad de software.

Uno de los impactos más importantes esperados, son las mejoras dentro de los procesos afectados e incorporados en la PYME seleccionada, los cuales son a largo plazo (no evaluado en este proyecto), dentro de los cuales se tiene:

Impacto Esperado	Plazo (años)	Indicador verificable	Supuestos
Mejora en los procesos actuales de desarrollo	1 - 2 Años	Evaluación del estado de la empresa	Continuar con el proceso de desarrollo definido
Mejora en la calidad del producto	1 - 2 Años	Resultados de auditoría de producto mediante norma de calidad	Continuidad en el uso de la norma de calidad
Mejora de la satisfacción del cliente	4 - 9 Años	Encuestas de satisfacción al cliente	Otras áreas como soporte e implementación tiene lineamientos de calidad
Disminución de costos de calidad	4 - 9 Años	Indicadores de costos de calidad	Continuación del proceso definido de desarrollo y la norma de calidad seleccionada

Tabla 17. Impactos esperados a futuro por el proyecto

6. DESARROLLO

6.1. FASE DE DIAGNÓSTICO

La empresa seleccionada para el presente proyecto, es una organización desarrolladora de productos de software ERP, CRM, Educativos y otros productos especiales, con clientes a nivel local y nacional, el principal producto comercializado es la aplicación ERP en la cual se encuentra su mayor porcentaje de clientes, actualmente posee algunos problemas de calidad del producto y de organización de los procesos, para lo cual la dirección ha iniciado el apoyo al fortalecimiento de estos procesos que permitan brindar un mejor servicio y producto a sus clientes, y de esta manera lograr una mejor imagen ante los mismos.

6.1.1. DESCRIPCIÓN PROCESO ACTUAL DE LA ORGANIZACIÓN

Actualmente la organización cuenta con un área de desarrollo del producto que emplea el marco de trabajo SCRUM para la gestión del desarrollo de software. Siguiendo los lineamientos de SCRUM, los roles que actualmente maneja la organización son:

Rol en SCRUM	Cargo en la Organización
<i>Scrum Master</i>	Líder de desarrollo y miembro del equipo
Propietario del producto (<i>Product owner</i>)	Gerente de la empresa
Equipo Scrum	Equipo de desarrollo

Tabla 18. Roles Actuales de la Organización
Fuente: Elaboración propia

Además de SCRUM la organización también trabaja de manera parcial con el modelo de procesos PSP 0, para el registro de los tiempos e interrupciones durante el desarrollo de las tareas. A continuación se presenta un diagrama donde se representan las actividades actualmente desarrolladas siguiendo el marco de trabajo de SCRUM junto con PSP.

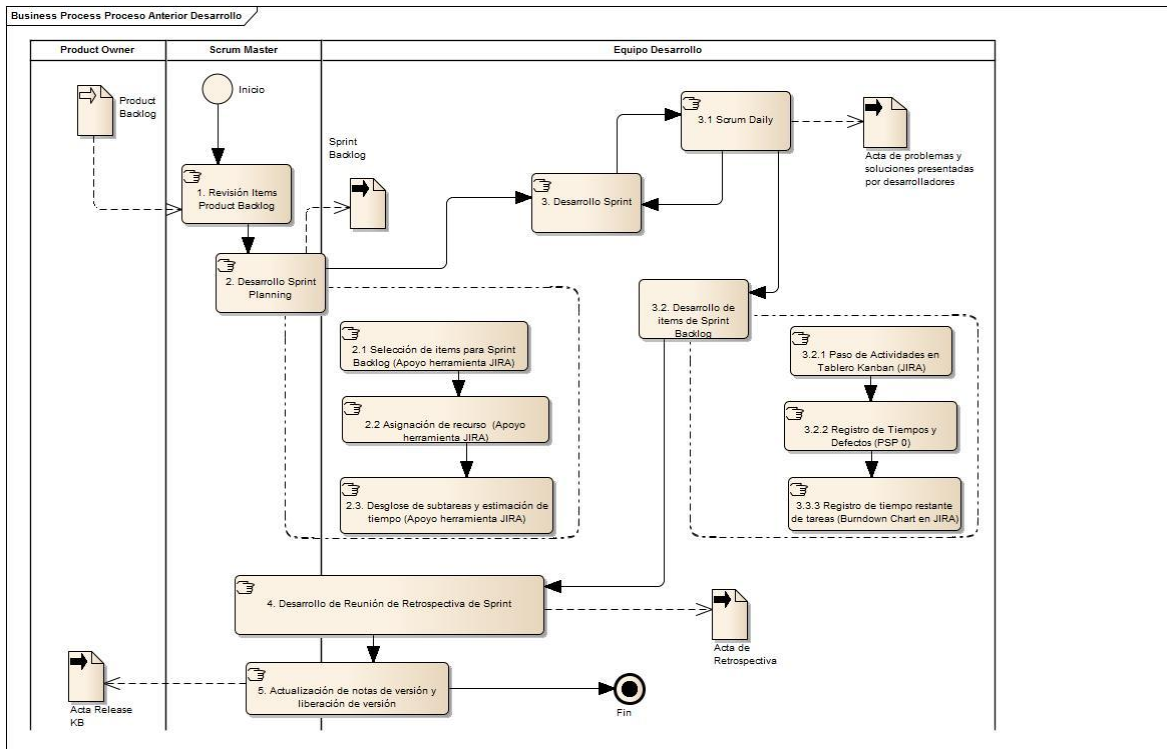


Figura 15. Diagrama de Actividades de SCRUM y PSP en la Organización actualmente
Fuente: Elaboración propia

Como se puede observar en el diagrama, como apoyo para el desarrollo de las diferentes actividades, se utilizan herramientas como JIRA que permite registrar las actividades en el ciclo de vida del proceso, además de otras herramientas desarrolladas por la organización para registrar los tiempos, interrupciones y defectos que se establecen en PSP.

6.1.2. HERRAMIENTAS USADAS

Como se mencionó anteriormente la organización posee una serie de herramientas para la gestión del proceso de desarrollo de software las cuales se nombran a continuación:

Herramienta	Proveedor	Descripción de Funcionalidad
JIRA	Atlassian	<ul style="list-style-type: none"> Aplicación donde se ingresan los reportes de errores y requerimientos (incidencias) por parte del equipo de soporte e implementación (Independientes del área de desarrollo). Manejo de las versiones y notas de liberación de los productos. Gestión de los <i>Sprints</i>, <i>ProductBacklog</i>, y Reportes de Scrum. Gestión de los diversos proyectos

		(Productos).
FishEye/Crucible	Atlassian	<ul style="list-style-type: none"> Herramienta de gestión de control de cambios de código. Ingreso de revisiones de código. Control de Cambios por proyecto (Producto-Integrada con Git)
Git	Git	<ul style="list-style-type: none"> Herramienta de repositorio y control de código
PSP propietario	Propia	<ul style="list-style-type: none"> Herramienta de registro de tiempos de desarrollo e interrupciones.

Tabla 19. Herramientas utilizadas en la organización
Fuente: Elaboración propia

6.1.3. VALORES DE MEDICIONES ACTUALES

Actualmente en el proceso de SCRUM, al realizar el *sprint planning* cada desarrollador realiza una estimación del tiempo que puede tardar en desarrollar la tarea asignada, de acuerdo a la explicación dada de la tarea, una vez inicia con la tarea se comienzan a registrar los tiempos de trabajo de la tarea con la herramienta de PSP propietaria, actualmente en su gran mayoría solo se registra la etapa de codificación y no las otras etapas de desarrollo como se describe en PSP, a continuación se presenta una tabla con los resultados de estas estimaciones versus los tiempos registrados:

Proyecto	Tiempo Estimado (Hrs)	Tiempo Real (Hrs)	Diferencia	Tareas Estimadas	Tareas con Registro Tiempo	Tareas Sin Registro Tiempo
SW-ERP G i4	700.33	841.86	141.53	209	144	65
SW-Actualizacion Gi4	26	26.16	0.16	3	3	0
SW-Asistencia v2	0.5	0	0.5	1	0	1
SW-Business-Ingelligence	12	12.5	0.5	1	1	0
SW-Educativo Web	6	1.88	4.12	2	2	0
SW-ERP Comercial v2	9.16	7.4	1.76	4	4	0
SW-ERP Contabilidad v2	2	1.83	0.17	1	1	0
SW-ERP Web	56	96.88	40.88	3	3	0
SW-Special Software (NT)	48	23.61	24.39	6	5	1
SW-Interoperabilidad	4	0	4	1	0	1
SW-Sincronizacion-ERP	9.5	7.26	2.24	2	1	1
Total	873.49	1019.4	220.25	233	164	69
Promedio	79.4	92.7	20.0			
Máximo	700.3	841.9	141.5			
Mínimo	0.5	0.0	0.2			
Mediana	9.5	7.4	2.2			

Tabla 20. Registro de Estimación de tiempo vs Real por Proyecto Fuente: Elaboración propia

Sprint	Tiempo Estimado (Hrs)	Tiempo Real (Hrs)	Diferencia	Total Tareas Estimadas	Total Tareas con Registro Tiempo	Total Tareas Sin Registro Tiempo
Sprint No 1	89.25	67.4	21.85	42	22	20
Sprint No 2	89.5	108.18	18.68	16	15	1
Sprint No 3	43.25	103.31	60.06	12	12	0
Sprint No 4	121.5	114.55	6.95	17	17	0
Sprint No 5	36	68.33	32.33	10	10	0
Sprint No 6	85	67.53	17.47	9	9	0
Sprint No 7	106.83	140.75	33.92	14	16	-2
Sprint No 8	55	27.35	27.65	4	4	0
Sprint No 9	118	130.2	12.2	13	13	0
Sprint No 10	93	142.66	49.66	11	11	0
Sprint No 11	26.5	20.16	6.34	5	5	0
Total	863.83	990.42	287.11	153	134	19
Promedio	78.5	90.0	26.1	13.9	12.2	1.7
Máximo	121.5	142.7	60.1	42.0	22.0	20.0
Mínimo	26.5	20.2	6.3	4.0	4.0	-2.0
Mediana	89.3	103.3	21.9	12.0	12.0	0.0

Tabla 21. Registro de Estimación de tiempo vs Real por Sprint
Fuente: Elaboración propia

Sprint	Total Tareas Programadas	Total de Tareas Completadas	Tareas Incompletas	% Completas	% Incompletas
Sprint No 1	51	42	9	82.40%	17.60%
Sprint No 2	22	17	5	77.30%	22.70%
Sprint No 3	24	11	13	45.80%	54.20%
Sprint No 4	24	9	15	37.50%	62.50%
Sprint No 5	21	14	7	66.70%	33.30%
Sprint No 6	16	13	3	81.30%	18.80%
Sprint No 7	22	14	8	63.60%	36.40%
Sprint No 8	17	11	6	64.70%	35.30%
Sprint No 9	22	15	7	68.20%	31.80%
Sprint No 10	21	14	7	66.70%	33.30%
Sprint No 11	19	13	6	68.40%	31.60%
Total	259	173	86	66.80%	33.20%
Promedio	23.5	15.7	7.8		
Máximo	51.0	42.0	15.0		
Mínimo	16.0	9.0	3.0		
Mediana	22.0	14.0	7.0		

Tabla 22. Registro de Número de Tareas Planeadas por Sprint
Fuente: Elaboración propia

RESUMEN REGISTRO DE TIEMPOS DETALLADO	
Total de Tareas	209
Total de Tareas Estimadas	178
Total de Tareas no Estimadas	31
Total de Tareas con registro de tiempos	137
Total de Tareas sin registro de tiempos	72

Tabla 23. Resumen de tabla de registro de tiempos de tareas, tomado del Anexo 1 (Registro de Estimación de tiempo vs Real por Tarea) Fuente: Elaboración propia

Las tablas anteriores son una muestra de las estimaciones y tiempo trabajado que el equipo de desarrollo ha reportado en las diferentes tareas ejecutadas en los proyectos que se manejan actualmente en la organización, de lo cual se puede concluir lo siguiente:

- En la tabla 19 Y 20 se observa que la suma de las diferencias de los tiempos entre lo estimado y las horas reales es del 25.21% y 29.61% con respecto a la suma total de lo estimado, lo cual hace ver que se requiere mejorar el proceso de estimación, ya que el proceso actual es intuitivo.
- El 33.2 % del total de las tareas programadas en los *sprints* no se completa, como se observa en la tabla 21, resultado reflejado de la falta de exactitud en la estimación.
- En la tabla 22 se evidencia que no hay un 100% de registro de tiempos de trabajo, ampliando la diferencia de lo estimado con respecto al tiempo registrado, este porcentaje corresponde al 12.41% del total de tareas.

6.1.3.1. CANTIDAD DE DEFECTOS REPORTADOS POR VERSIÓN

A continuación se relaciona la tabla que contiene la información de los registros de errores encontrados en la aplicación principal de la organización (Business Management), por el equipo de implementación y soporte durante cada liberación de las respectivas versiones (esto corresponde a errores encontrados por el cliente), el registro de la versión se establece de acuerdo a la versión que se encuentra instalada en el momento de presentarse el error, esto quiere decir que no necesariamente un error corresponde exactamente a una funcionalidad liberada en la versión.

Versión	No Defectos	% Sobre Total
3.7.x	38	5.14%
3.8.x	24	3.25%
3.9.x	74	10.01%
3.10.x	66	8.93%

3.11.x	89	12.04%
3.12.x	67	9.07%
3.13.x	107	14.48%
3.14.x	147	19.89%
3.15.x	31	4.19%
3.16.x	59	7.98%
3.17.x	37	5.01%
TOTAL	739	100.00%

Tabla 24. Registro de errores de acuerdo a versión reportada de la aplicación

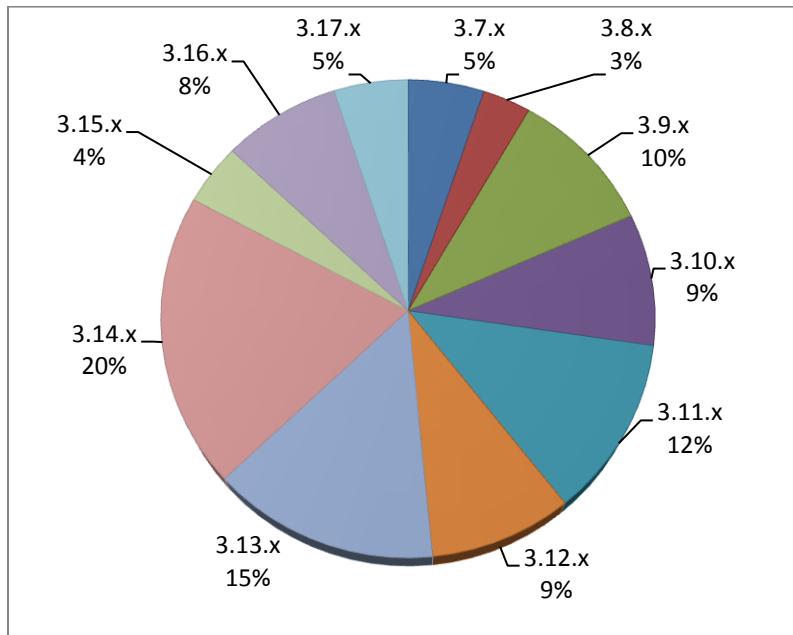


Figura 16. Gráfico de registro de errores de acuerdo a versión reportada de la aplicación

Se puede encontrar que existe un alto reporte de errores durante las diferentes versiones liberadas en la aplicación en promedio el 9.09%, lo cual permite establecer que se requiere mejorar el proceso de control de calidad del producto, con el fin de disminuir estos reportes y costos de corrección de errores detectados externamente.

6.1.3.2. ENCUESTA DE PERCEPCIÓN DEL PRODUCTO

Para conocer la percepción que actualmente tienen los clientes sobre el producto, de tal manera que permita dar otra dimensión al estado actual de la organización respecto a sus procesos y nivel de calidad, se realiza la siguiente encuesta:

Ficha Técnica Encuesta	
Nombre del Estudio	Encuesta de evaluación del producto
Liderada por :	Jhon Alexander Holguín Barrera
Ejecutada y analizada por:	Jhon Alexander Holguín Barrera
Instrumento:	Cuestionario de 10 Preguntas
Marco de la muestra:	Usuarios de la aplicación de empresas actualmente poseen los productos de la empresa
Metodología:	Cuantitativa
Muestreo:	Aleatorio Simple
Técnica:	Encuesta virtual a través de la aplicación Survey Monkey ²
Fecha de realización:	22 de Agosto de 2013
Tamaño de la muestra:	49 Personas

Tabla 25. Ficha técnica encuesta "Evaluación de calidad del producto"

PREGUNTAS Y RESULTADOS DE LA ENCUESTA

1. ¿Hace cuánto hace uso de nuestros productos?

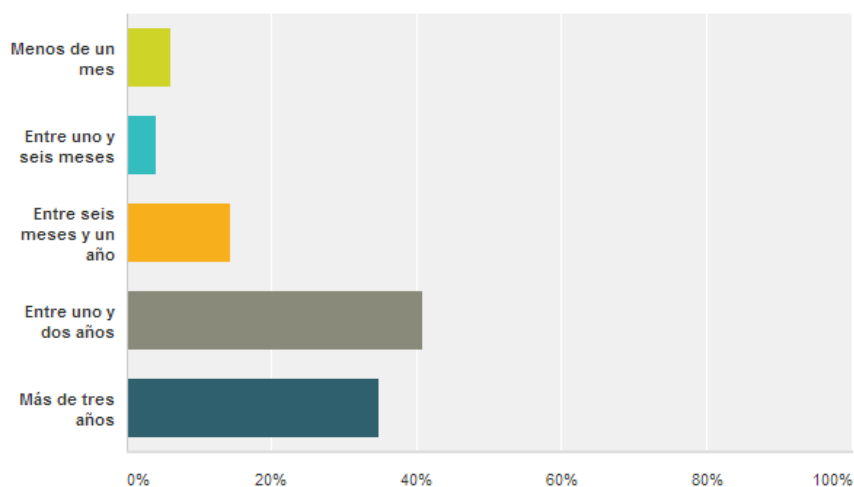


Figura 17. Resultados, ¿Hace cuánto hace uso de nuestros productos?

² www.surveymonkey.com: Herramienta de encuestas en línea.

2. Seleccione las aplicaciones que su organización maneja con la organización

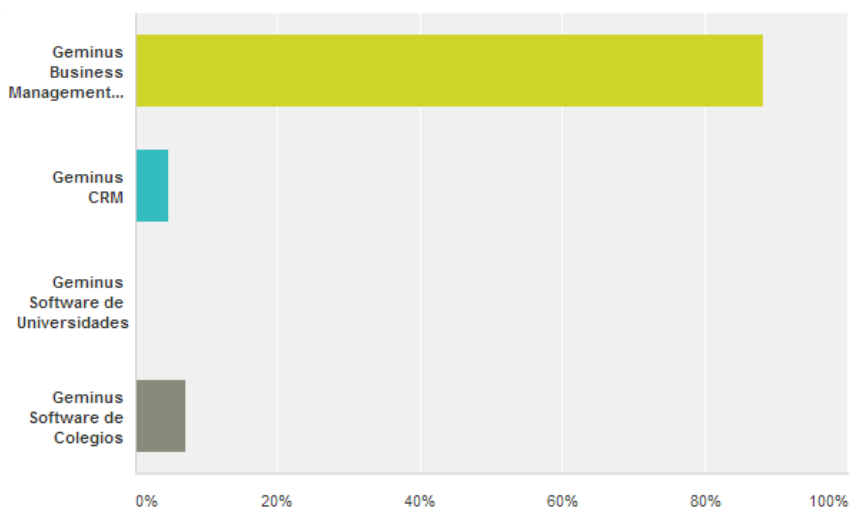


Figura 18. Resultados, Seleccione las aplicaciones que su organización maneja con la organización

3. Responda esta respuesta si seleccionó Business Management (ERP) en la respuesta anterior, ¿Cuáles procesos de la aplicación tiene implementados en su organización?

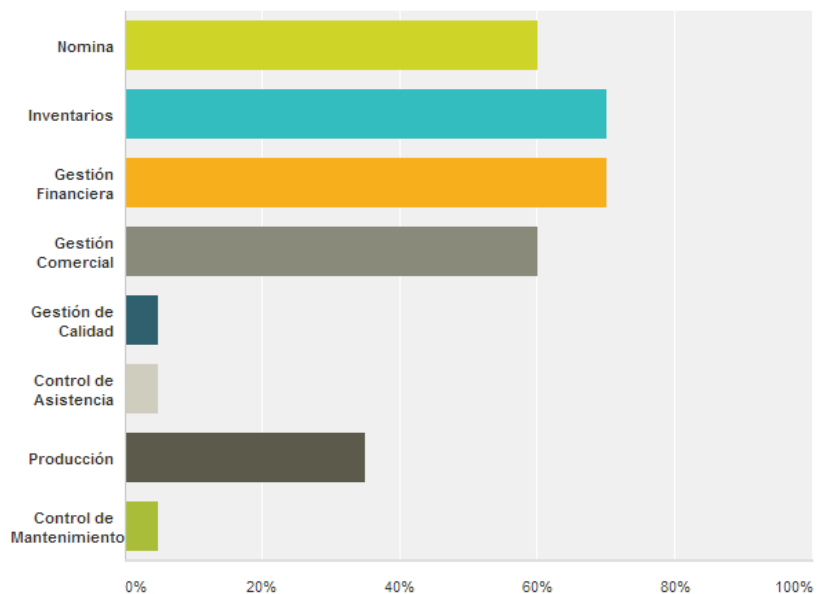


Figura 19. Resultados, ¿Cuáles procesos de la aplicación tiene implementados en su organización?

4. En general, ¿considera que las aplicaciones de la organización cumplen con las necesidades de su organización?

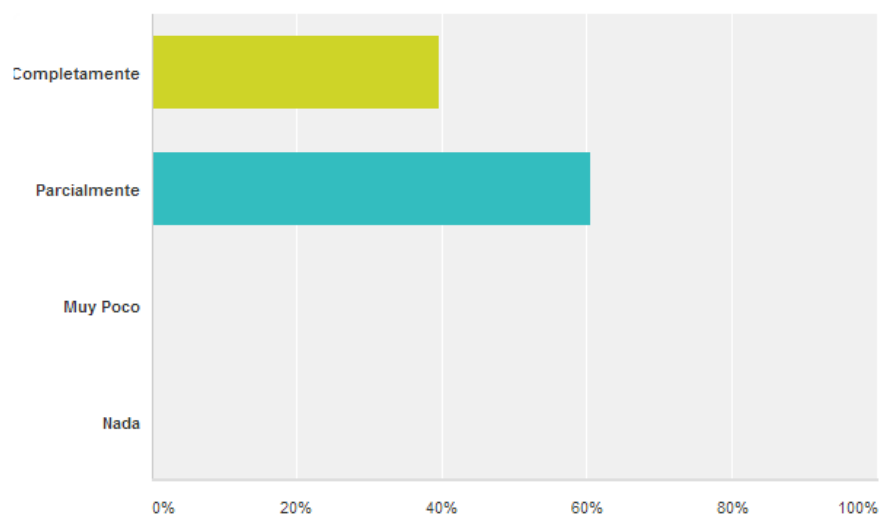


Figura 20. Resultados, ¿considera que las aplicaciones de la organización cumplen con las necesidades de su organización?

5. En general cómo califica la calidad de las aplicaciones que su organización maneja con la organización

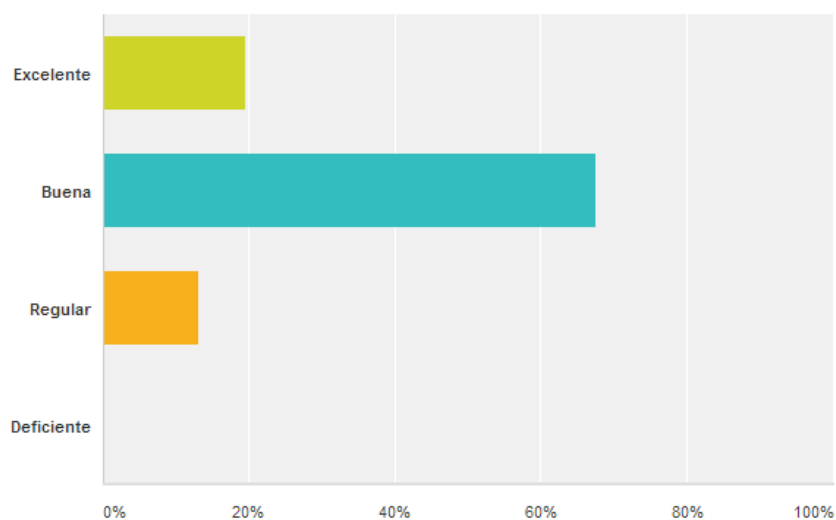


Figura 21. Resultados, En general como califica la calidad de las aplicaciones que su organización maneja con la organización.

6. En general, ¿Cada cuánto recibe actualizaciones de las aplicaciones de la organización?

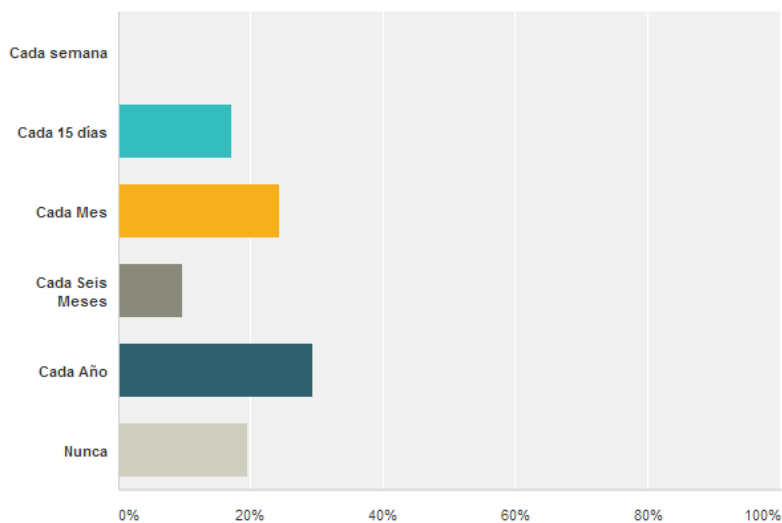


Figura 22. Resultados, ¿Cada cuánto recibe actualizaciones de las aplicaciones de la organización?

7. Cuando recibe actualizaciones de las aplicaciones de la organización, por lo general la actualización:

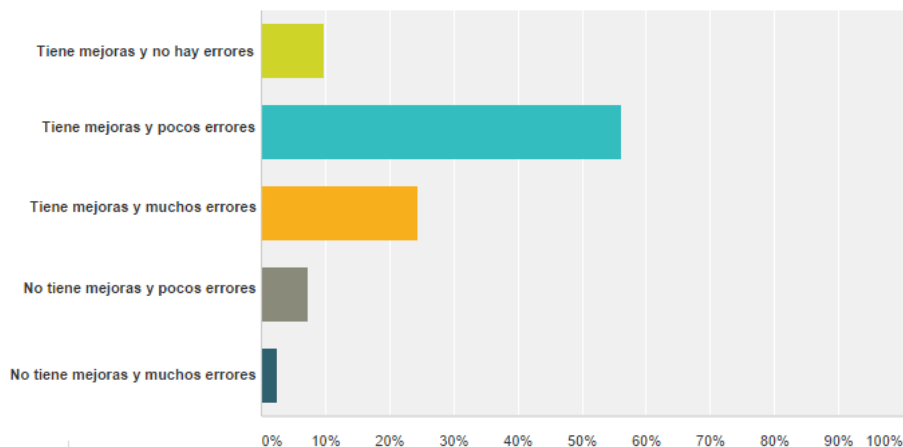


Figura 23. Resultados, Cuando recibe actualizaciones de las aplicaciones de la organización, por lo general la actualización.

8. Cuando reporta un error de las aplicaciones de la organización, ¿cuánto debe esperar para que se corrija este error?

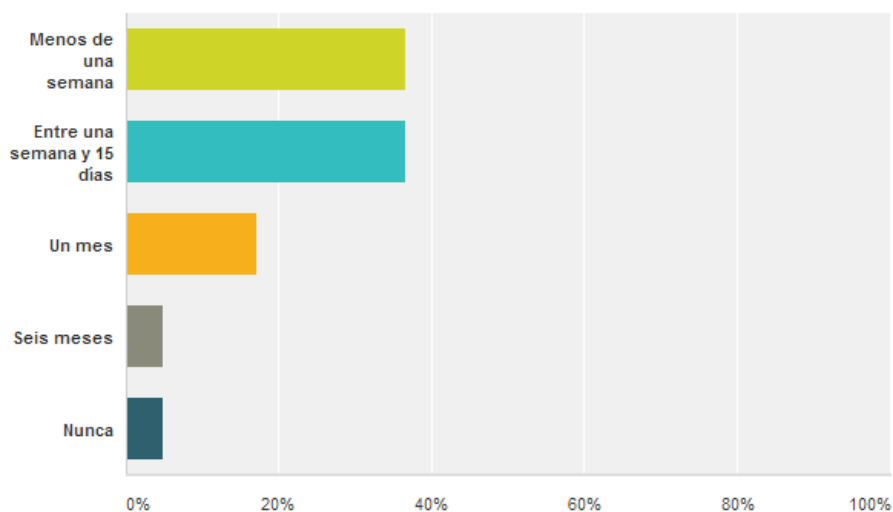


Figura 24. Resultados, Cuando reporta un error de las aplicaciones de la organización, ¿cuánto debe esperar para que se corrija este error?

9. Cuando recibe la corrección a un error que reportó de las aplicaciones de la organización, por lo general:

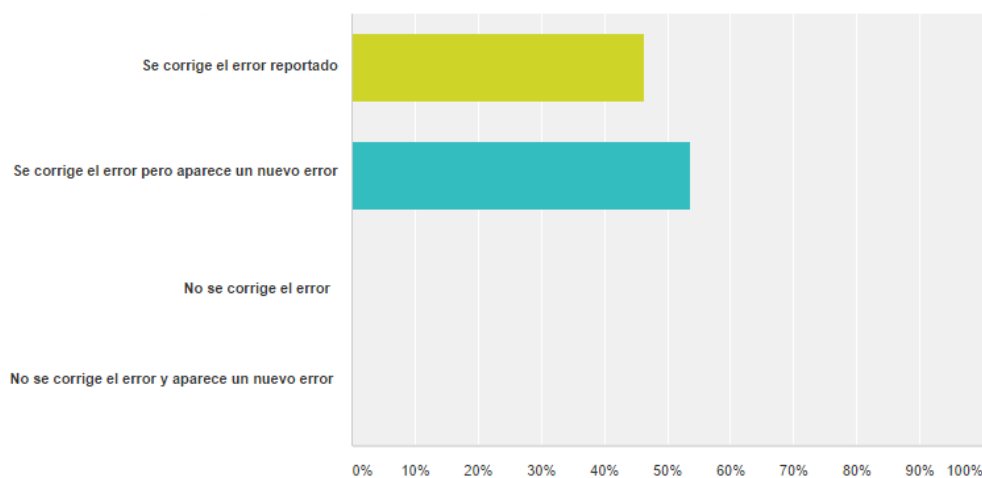


Figura 25. Resultados, Cuando recibe la corrección a un error que reportó de las aplicaciones de la organización, por lo general:

10. ¿Recomendaría las aplicaciones de la organización a otras empresas?

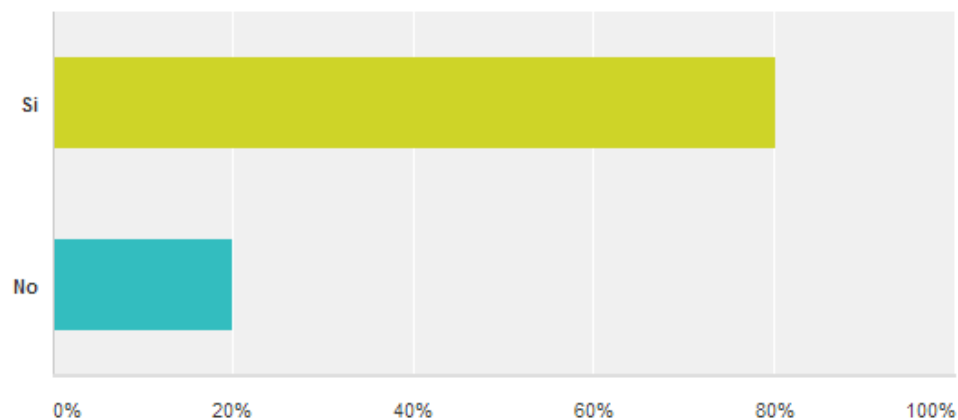


Figura 26. Resultados, ¿Recomendaría las aplicaciones de la organización a otras empresas?

De acuerdo a los resultados de la encuesta, se puede observar que las preguntas 7, 8 y 9 que corresponden a preguntas en donde se busca obtener la percepción de los clientes frente a las diferentes inconsistencias que perciben en la aplicación desde que reciben una actualización con mejoras, nuevas funcionalidades o correcciones; reflejan que existen falencias en la calidad detectados por los clientes, dentro de las cuales se puede resaltar la pregunta 9 donde los clientes perciben que aunque pueden recibir correcciones a problemas ya reportados, aparecen nuevos problemas, lo cual puede convertirse en un ciclo continuo en el cual nunca se da una solución contundente y eficaz al cliente, por otro lado estos problemas de calidad se convierten en costos externos, los cuales pueden llegar a ser los de mayor repercusión negativa en la imagen de la organización; para esto se hace necesario establecer un plan de calidad que ayude a desarrollar un producto de calidad y evite que los usuarios del producto detecten este tipo de defectos en el uso.

6.1.3.3. ENCUESTA DE PERCEPCIÓN DEL EQUIPO

Otro de los puntos de vista importantes para tener en cuenta del estado actual de la organización, es conocer la percepción que tiene el equipo acerca del proceso de desarrollo y de la calidad del producto, para esto se llevó a cabo la encuesta que se presenta a continuación como un elemento que permita obtener la opinión del equipo:

Ficha Técnica Encuesta	
Nombre del Estudio	Encuesta equipo de desarrollo
Liderada por :	Jhon Alexander Holguín Barrera
Ejecutada y analizada por:	Jhon Alexander Holguín Barrera
Instrumento:	Cuestionario de 10 Preguntas
Marco de la muestra:	Equipo de desarrollo de la organización
Metodología:	Cuantitativa
Muestreo:	Todo el equipo
Técnica:	Encuesta virtual a través de la aplicación Survey Monkey
Fecha de realización:	26 de Agosto de 2013
Tamaño de la muestra:	6 Personas

Tabla 26. Ficha técnica encuesta "Percepción del proceso del equipo de desarrollo"

PREGUNTAS Y RESULTADOS DE LA ENCUESTA

1. ¿Considera que actualmente existe un proceso definido de desarrollo de software dentro de la organización?

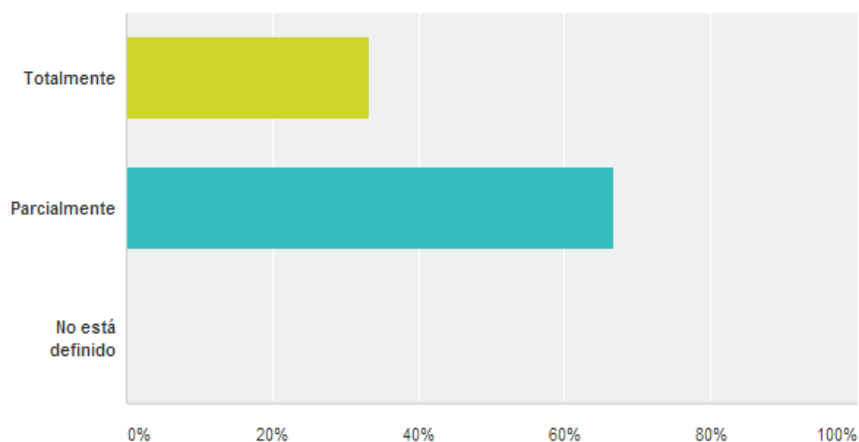


Figura 27. Resultados pregunta, ¿Considera que actualmente existe un proceso definido de desarrollo de software dentro de la organización?

2. ¿Considera que posee la suficiente capacitación de la aplicación del marco de trabajo de SCRUM dentro de la organización?

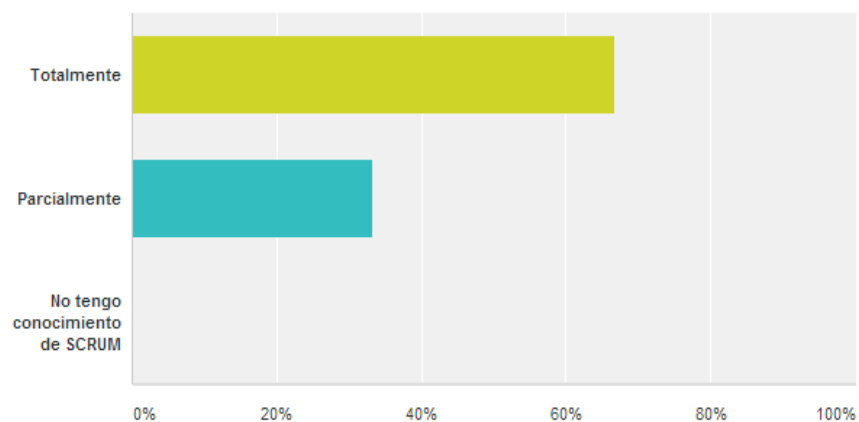


Figura 28. Resultados pregunta, ¿Considera que posee la suficiente capacitación de la aplicación del marco de trabajo de SCRUM dentro de la organización?

3. ¿Considera que posee la suficiente capacitación de la aplicación de PSP dentro de su proceso de desarrollo de software?

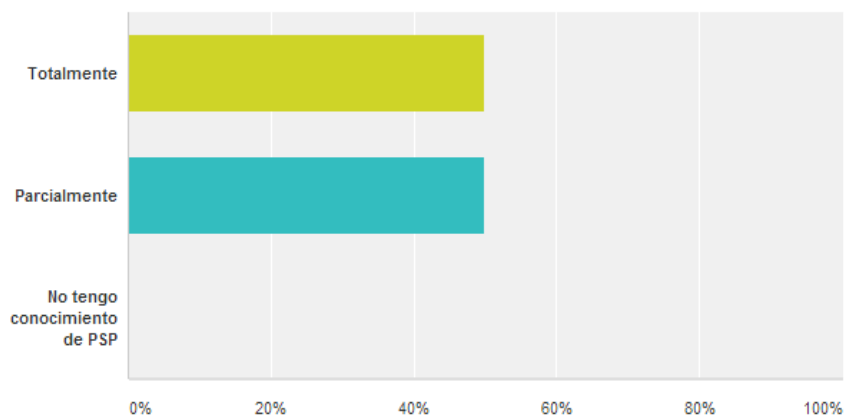


Figura 29. Resultados pregunta, ¿Considera que posee la suficiente capacitación de la aplicación de PSP dentro de su proceso de desarrollo de software?

4. ¿Considera que el marco de trabajo SCRUM y el modelo de proceso PSP están integrados dentro de un proceso definido de desarrollo dentro de la organización?

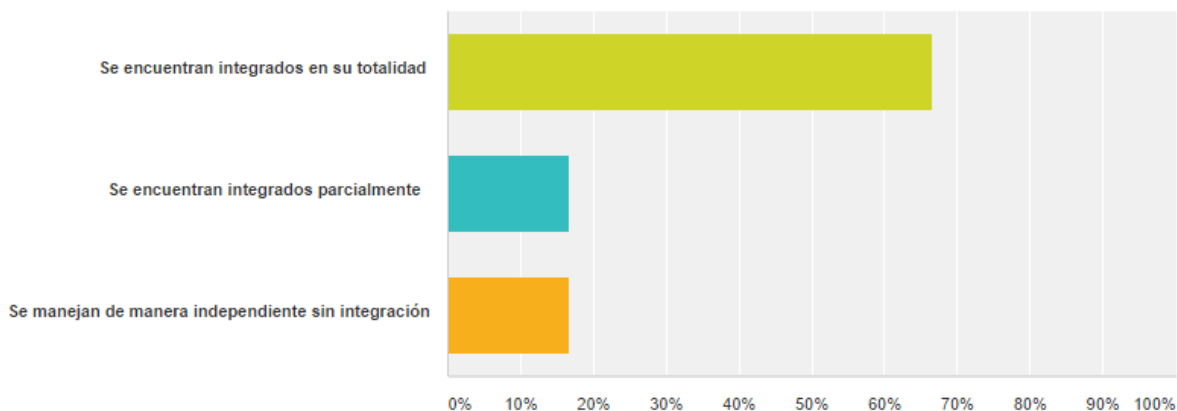


Figura 30. Resultados pregunta, ¿Considera que el marco de trabajo SCRUM y el modelo de proceso PSP están integrados dentro de un proceso definido de desarrollo dentro de la organización?

5. ¿Considera que el proceso de desarrollo esta guiado por un modelo de control de calidad estándar definido?

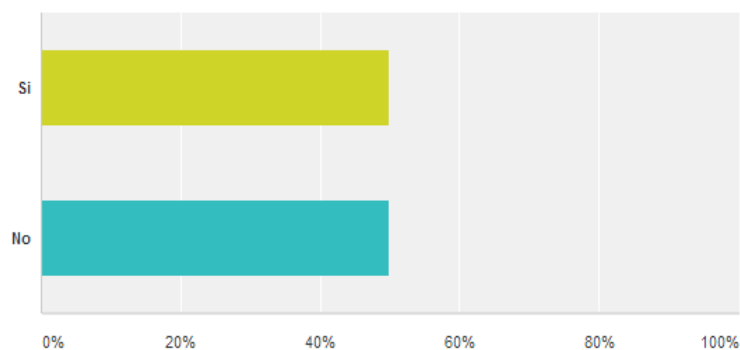


Figura 31. Resultados pregunta, ¿Considera que el proceso de desarrollo esta guiado por un modelo de control de calidad estándar definido?

6. ¿Considera que existe la suficiente documentación para la ejecución de las actividades de desarrollo asignadas?

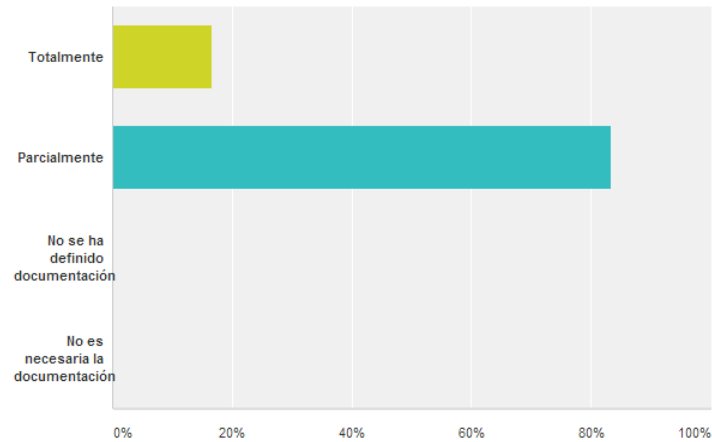


Figura 32. Resultados pregunta, ¿Considera que existe la suficiente documentación para la ejecución de las actividades de desarrollo asignadas?

7. Considera que la estimación de las tareas de desarrollo dentro de los proyectos son:

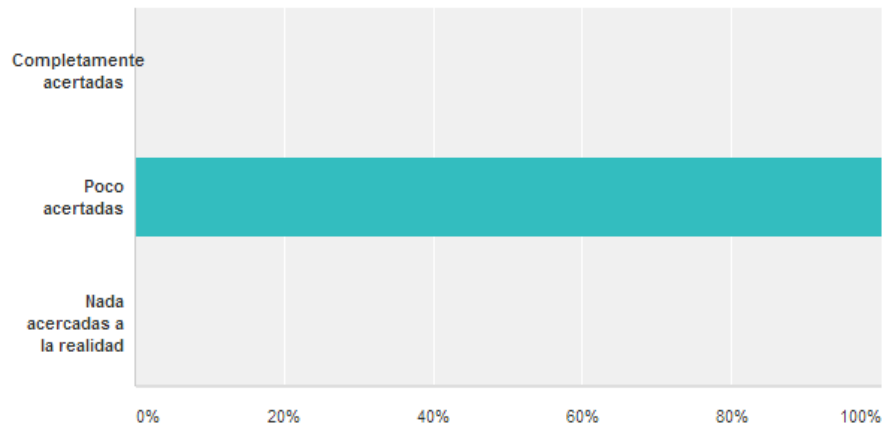


Figura 33. Resultados pregunta, Considera que la estimación de las tareas de desarrollo dentro de los proyectos son.

8. ¿Considera la aplicación de SCRUM y PSP dentro del proceso de desarrollo, favorece al ambiente de trabajo del equipo?

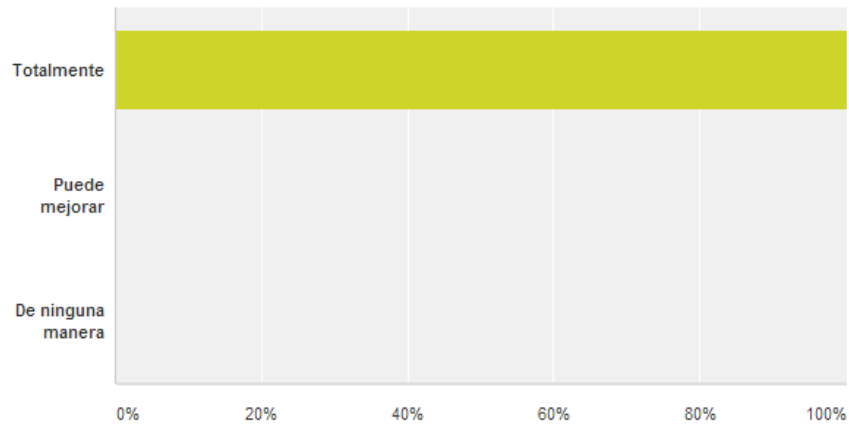


Figura 34. Resultados pregunta, ¿Considera la aplicación de SCRUM y PSP dentro del proceso de desarrollo, favorece al ambiente de trabajo del equipo?

9. ¿Cuál es la percepción de calidad que posee de los productos desarrollados por la organización?

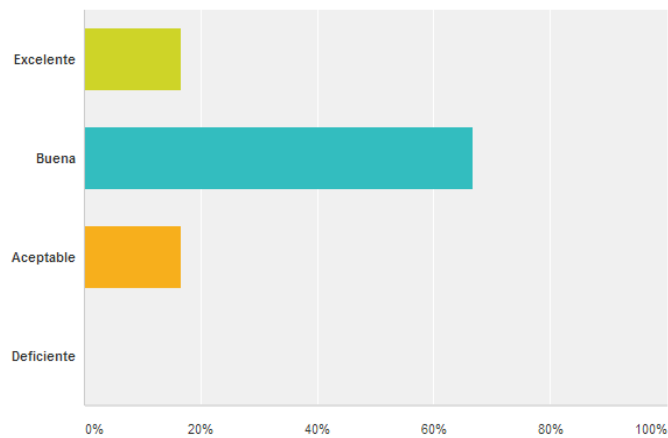


Figura 35. Resultados pregunta, ¿Cuál es la percepción de calidad que posee de los productos desarrollados por la organización?

10. ¿Qué recomendaciones puede dar para mejorar el proceso de desarrollo que actualmente se maneja en la organización?

RESPUESTAS
-Mejorar en cuanto a documentación.
Lo más importante son las pruebas externas y que exista una retroalimentación
Sabemos que estimar el tiempo para el desarrollo es importante, pero debido a factores externos siempre es

difícil cumplir con los tiempos estimados.
Que se pueda dedicar más tiempo a la implementación de los ítems de una metodología de desarrollo, y que no se exija terminar todas las tareas en un tiempo muy corto, porque esto no deja realizar bien la práctica de SCRUM y PSP
Formalizar el proceso de desarrollo e integrarlo, de igual forma establecer dentro del proceso pruebas, y un control de calidad

De los resultados de la anterior encuesta al equipo de desarrollo se puede rescatar los siguientes elementos:

- De la pregunta número uno, se obtiene que en la organización no existe un proceso claramente definido, lo cual es importante para que el equipo pueda aplicarlo a cabalidad.
- De la pregunta número 4 la mayoría del equipo percibe que SCRUM y PSP están integrados, no obstante esta percepción está concebida debido a la poca capacitación del proceso de PSP que se evidencia en la pregunta 3.
- De la pregunta 7 se resalta que el equipo coincide que no existe una herramienta de estimación eficaz que permita realizar este proceso de manera acertado, para lo cual PSP es un componente clave que puede apoyar este proceso.
- De la pregunta 8 otro elemento a resaltar es que SCRUM y PSP favorecen al ánimo del equipo, lo cual puede convertirse en una ventaja que permita implementar el proyecto de una manera más eficaz.

6.1.4. ACTIVIDADES CUMPLIDAS EN SCRUM Y PSP

A continuación se relacionan las actividades que realizan tanto del marco de trabajo SCRUM como PSP, en la organización:

SCRUM		
Actividad	Cumple (si/no/parcial)	Observaciones
<i>Sprint Planning</i>	Si	La estimación de tiempo de las tareas se realiza a través de la evaluación de la complejidad de las tareas por <i>Planning Pocker</i> y el tiempo es establecido por el desarrollador.

<i>ScrumDaily</i>	Parcial	Existe la falencia de cumplir las reuniones diarias, ya que no se realizan siempre por atender solución a problemas reportados.
<i>Sprint Review</i>	No	
<i>Sprint Retrospective</i>	Si	Se realiza la reunión con el equipo donde se establece lo que estuvo bien, por mejorar y lo que se quiere probar para el siguiente <i>sprint</i>
PSP 0		
Planeación	No	
(Desarrollo) Diseño	No	
Codificación	Si	Se codifican las tareas establecidas en el sprint planning guiados por los prototipos o documentación existente, Aunque se ejecuta esta actividad, solo se registran los tiempos de desarrollo y no de las demás fases (Pruebas, Diseño, etc.)
Compilación	No	
Pruebas	Si	Se realizan a nivel individual por cada desarrollador, pero no se lleva registro de las mismas.
<i>Postmortem</i>	No	
Registro de tiempos	Si (Parcial)	Se realiza registro de tiempos de las tareas pero no se cumplen todas las fases definidas en PSP, por lo general se registran las fases de diseño y codificación.
Registro de defectos	No	

Tabla 27. Actividades de SCRUM y PSP, actualmente realizadas en la organización

Se puede observar que las actividades tanto de SCRUM como PSP, actualmente no se está realizando de manera completa, ya que no se posee un proceso integrado entre estos, claramente definido.

6.1.5. FORTALEZAS Y PUNTOS DE MEJORA DEL PROCESO ACTUAL

Fortalezas

- Existe un apoyo de la dirección de la empresa para la implementación de metodologías que permitan la mejora de la calidad de los productos de la organización.
- Dentro del equipo de desarrollo ya se han empezado a utilizar los marcos de procesos SCRUM y PSP, aunque parcialmente. Existe un ambiente positivo en los miembros del equipo para implementar prácticas que permitan mejorar su productividad y

rendimiento, esto se puede evidenciar en los resultados de la encuesta al equipo desarrollador en la pregunta número 8.

- Existen herramientas que facilitan la implementación de estos marcos de trabajo, las cuales pueden ser ajustadas a las necesidades del equipo.
 - JIRA (Gestión del ciclo de vida del producto y ciclo de SCRUM)
 - FISHEYE (Control para revisiones de código)
 - PSP Propietaria (Toma de tiempos, defectos e interrupciones automático)
 - GIT (Herramienta de control de código)

Puntos de Mejora

- Se deben mejorar el cumplimiento de la ejecución de las actividades establecidas en SCRUM y PSP para hacer efectivos los resultados de las metodologías.
- Se deben establecer estrategias que permitan mejorar la estimación de las tareas.
- Se debe establecer un modelo de control de calidad que permita filtrar los posibles errores que se liberen en los desarrollos ejecutados, esto se puede hacer a través de pruebas más formales de los desarrollos ejecutados por los desarrolladores, con herramientas, revisiones por pares u otras técnicas de control de calidad.
- No existen actividades de diseño de las tareas a desarrollar, que permitan a los desarrolladores tener claro el trabajo que tiene que realizar.

6.1.6. SELECCIÓN DE MODELO DE CALIDAD

Para la evaluación de las diferentes normas seleccionadas se utilizó la siguiente tabla que permite establecer si se requieren o no las características seleccionadas, que a su vez permiten definir cuál es la norma que se utilizará como modelo de evaluación de calidad del producto de software:

Valor Puntuación	Descripción
R	Define que dentro de la norma se requiere el ítem o característica.
NR	Define que dentro de la norma no está definida como requerida el cumplimiento de la característica

Tabla 28. Escala evaluación modelo de calidad

Característica Global	Característica	Descripción	ISO 25000	ISO 9126	ISO 14598
Consideraciones funcionales	Atributos no funcionales	Corresponde a la evaluación de las características que no corresponden a funcionalidades de la aplicación si no restricciones o requerimientos de calidad (IEEE, Computer Society, 2004).	R	R	NR
	Atributos funcionales	Corresponde a la evaluación de las funcionalidades específicas que debe cumplir el sistema. (IEEE, Computer Society, 2004)	R	R	NR
Generales	Vigencia	Establece si la norma esta actualmente vigente o si ya es obsoleta.	Vigente	En reemplazo o	En Reemplazo
	Organismos que soportan la norma	Que organismos internacionales soportan o patrocinan la norma	ISO	ISO	ISO
	Permite certificación	Valida si el cumplimiento de la norma puede certificarse ante un ente regulador autorizado.	SI	NO	NO
	Establece métricas de evaluación	Permite establecer si la norma describe métricas que permitan ser guía para validar el cumplimiento de la norma.	SI	SI	NO

Tabla 29. Calificación cualitativa para selección de modelo de calidad

Según el cuadro anterior se selecciona la norma ISO 25000 para la evaluación del producto debido a que es la norma vigente en el momento, contrario a las otras que están siendo reemplazadas, por otro lado evalúa características funcionales y no funcionales del producto.

Debido a lo amplio de esta norma se procederá a seleccionar la división y características más relevantes para el presente proyecto (Tabla 29), de acuerdo con las características propias de la empresa, el equipo de desarrollo y las aplicaciones desarrolladas; para esta evaluación se tuvo en cuenta la siguiente tabla donde se desglosan las divisiones:

DIVISIÓN	DESCRIPCIÓN	SEL	OBSERVACIONES
2501n	Se presentan modelos de calidad detallados incluyendo características para calidad interna y externa y en uso del producto de software		
2502n	Incluye un modelo de referencia de la medición de la calidad del producto, estableciendo medidas de calidad y guías para su aplicación.	X	Esta división establece de manera más explícita modelos de medición de la calidad del producto.

2503n	Permite especificar los requisitos de calidad que sirven como base de la obtención de requisitos del producto de software a desarrollar o como entrada del proceso de evaluación		
2504n	Proporciona recomendaciones, requisitos y guías para ejecutar el proceso de evaluación.		

Tabla 30. Divisiones norma ISO/IEC 25000

Descripción de la norma seleccionada ISO/IEC 2502n (25023)

La división de medición de la norma ISO/IEC 2502n se divide en varias normas subyacentes como se muestra en la figura 35, para el presente proyecto se ha seleccionado la norma ISO/IEC 25023 ya que este comprende de manera explícita las métricas de calidad para la evaluación de la calidad de un producto de software.

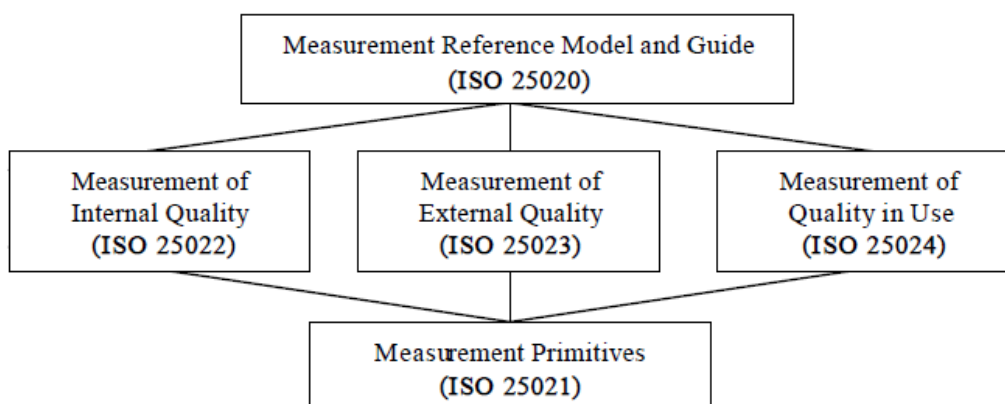


Figura 36. Estructura de la división de métricas de calidad, Fuente: An Information Model for Software Quality Measurement with ISO Standards p. 2 (Abran, Al Qutaish, Desharnais, & Habra, 2005)

Este estándar internacional provee un conjunto básico de métricas para la calidad de software o sistema a nivel interno y externo, las cuales pueden ser usadas con el modelo de calidad de la ISO/IEC 25010, la selección de estas métricas está a consideración y de acuerdo a las necesidades del usuario de la norma e incluso puede incluir métricas que no se encuentren establecidas (ISO/IEC, 2011); Las métricas establecidas en la norma están enfocadas a la calidad interna y externa, las cuales se encuentran fuertemente ligadas entre sí (Figura 36).

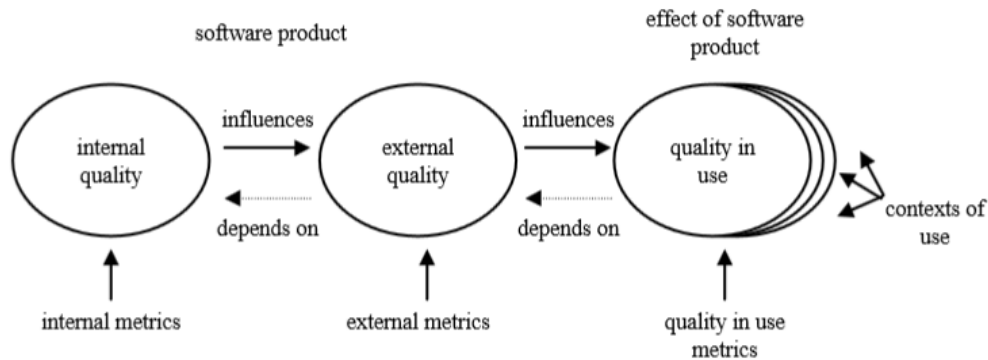


Figura 37. Relación entre las métricas de los tipos de calidad
Fuente ISO/IEC 25023, p. 14 (ISO/IEC, 2011)

Calidad Interna

Las métricas de la calidad interna, pueden ser aplicadas a un producto no ejecutable de software, durante sus etapas de desarrollo, esta provee a los usuarios la posibilidad de medir la calidad de los entregables intermedios, y así predecir la calidad del producto final, lo cual permite identificar y corregir de manera temprana lo que este mal durante el ciclo de desarrollo.

Calidad Externa

Las métricas de la calidad externa únicamente pueden ser usadas durante las etapas de pruebas (testing) del ciclo de vida del proceso y etapas operacionales.

Formato para documentar las métricas de calidad

Las métricas de calidad están basadas en las características y sub-características definidas en la ISO/IEC 25010 (Tabla 5), a su vez en la norma se definen los siguientes elementos a ser tenidos en cuenta para cada característica a evaluar (ISO/IEC, 2011):

- **Nombre:** La métrica de calidad posee un único nombre que puede ser identificado por un numero consecutivo si es necesario.
- **Descripción:** Expresa la pregunta a ser respondida a través de la aplicación de la métrica.
- **QMEs:** Elementos de la métrica de calidad relevantes que se utilizaron como entrada para la métrica de calidad.
- **Tipo de Métrica:** Los tipos usados son: Tamaño (e.g. Tamaño de la función, tamaño del código), Tiempo (e.g. Tiempo transcurrido, Tiempo del usuario), Conteo (e.g. Número de cambios, Número de fallas)

Con el fin de establecer los indicadores a utilizar para medir la calidad del producto, se utilizara el método GQM , basándose en los elementos definidos en la norma a tener en cuenta, para esto se realizó la siguiente tabla que mapea estos elementos:

Elemento Norma	Elemento GQM
Nombre	Meta
Descripción	Preguntas
QMEs	Métricas
Tipo Métrica	Tipo Métrica

Tabla 31. Mapeo de características norma ISO vs GQM Fuente: Elaboración propia

Definición de métricas de calidad seleccionadas

Debido al tamaño y extensión de esta norma, se seleccionaron las métricas más relevantes para el proyecto de acuerdo a las necesidades de la organización (Pardo, Pino, Garcia, & Piattini, 2009), también para las categorías de calidad se realizó una evaluación con el propietario del producto, al cual a través de una escala de 1 a 5 se le pidió que calificara el grado de importancia de la categoría para la organización según la definición realizada de estas categorías en el apartado donde se trata la norma, y complementando con una breve observación que explicara el porqué de la valoración. A continuación se muestran los resultados:

Categoría	Pertinencia (0-5)	Seleccionada	Observaciones <i>Product Owner</i>
Funcionalidad	5	X	Representa el todo de nuestra aplicación ya que es a través del correcto funcionamiento de las funcionalidades que el producto va a generar los resultados esperados por el cliente
Confiabilidad	3		Este es un requerimiento inmerso en las funcionalidades ya que en la definición del requerimiento o funcionalidad se establecen los parámetros de aceptación
Usabilidad	4	X	Uno de los objetivos de la organización es lograr que los usuarios aprendan a usar la aplicación de manera intuitiva, con el fin de que no dependan de una capacitación y además que puedan interactuar con las nuevas funcionalidades sin tener que leer un manual en lo posible
Rendimiento/Eficiencia	4	X	El grado de respuesta es muy importante para nuestros clientes, ya que se requiere una disponibilidad de la información en el momento que se requiere y de manera rápida, con el fin de apoyar en la toma de decisiones.
Mantenibilidad	4	X	Internamente la capacidad de que un desarrollador pueda modificar el código de otro desarrollador es demasiado importante ya que esto influye directamente en los costos que generan las modificaciones al software.
Portabilidad	2		Nuestras aplicaciones están diseñadas para entornos Windows y es nuestro nicho de mercado no nos interesa por el momento movernos a otras plataformas.
Compatibilidad	2		Nuestros sistemas no interactúan por lo general con otros sistemas
Seguridad	3		La seguridad y protección de los datos están dados por los parámetros de seguridad que tiene nuestra aplicación, por otro lado también depende de la gestión del DBA y director de sistemas de nuestros clientes.

Tabla 32. Selección de características relevantes de la norma a cumplir Fuente: Elaboración propia

CATEGORÍAS SELECCIONADAS

- Funcionalidad
- Rendimiento
- Usabilidad
- Mantenibilidad

De acuerdo a las categorías seleccionadas se utilizó el método GQM con el fin de establecer las métricas a evaluar alineadas a los indicadores de calidad establecidos en la norma para la medición, que permitirán realizar la medición para verificar la calidad del producto.

A continuación se presenta la definición de las métricas para cada categoría.

ABREVIATURA	DESCRIPCIÓN
C	CONTEO
P	PORCENTAJE

Tabla 33. Tipos de métricas establecidas para norma de calidad
Fuente: Elaboración propia

ABREVIATURA	DESCRIPCIÓN
I	INTERNA
E	EXTERNA

Tabla 34. Focos de métricas establecidas para norma de calidad
Fuente: Elaboración propia

FUNCIONALIDAD				
Meta	Preguntas	Métricas	Tipo Métrica	Enfoque Métrica
Evaluar el cumplimiento de los requerimientos definidos en un Sprint	¿Se implementaron todos los requerimientos en el Sprint?	Porcentaje de requerimientos se entregaron en el Sprint con respecto al total planeados	P	I
	¿Los requerimientos desarrollados cumplen su función?	Porcentaje de desarrollos cumplen con todas las expectativas descritas en el requerimiento	P	I
	¿Los requerimientos son exactos en la información presentada al usuario?	Porcentaje de requerimientos entregados en el Sprint tuvieron errores en datos o funcionamiento durante la fase de pruebas con respecto al total de requerimientos desarrollados	P	E

Tabla 35. Medidas de funcionalidad seleccionadas norma ISO 25023
Fuente: Elaboración propia

RENDIMIENTO				
Meta	Preguntas	QMEs	Tipo Métrica	Enfoque Métrica
Establecer la eficiencia de las funcionalidades implementadas en el Sprint	¿Qué tan eficiente son las funcionalidades desarrolladas en el Sprint?	Porcentaje de número de tareas del total que no sobrepasan el tiempo definido para completar el trabajo	P	I
	¿Cuánto procesamiento utiliza una funcionalidad para ser finalizada?	Porcentaje de tareas del total que no sobrepasan el Porcentaje de procesamiento definido desde el inicio a la finalización de una tarea	P	I
	¿Cuánto espacio de memoria es utilizado para finalizar una tarea dada?	Porcentaje de tareas del total que no sobrepasan el porcentaje de consumo de memoria definido desde el inicio de la tarea hasta la finalización	P	I

Tabla 36. Medidas de rendimiento seleccionadas norma ISO 25023
Fuente: Elaboración propia

USABILIDAD				
Meta	Preguntas	QMEs	Tipo Métrica	Enfoque Métrica
Establecer la usabilidad de las funcionalidades implementadas en el Sprint	¿Cuánto tiempo tardan los usuarios en aprender a usar una funcionalidad?	Porcentaje de tiempo adicional requerido del estimado para el aprendizaje de uso de la funcionalidad	T	E
	¿Qué proporción de funciones requieren demostración?	Porcentaje de funcionalidades implementadas sobre el total que requieren de una demostración	C	E
	¿Qué proporción de las funcionalidades son correctamente descritas en la documentación de usuario o ayuda?	Porcentaje de funcionalidades que poseen una documentación o ayuda de usuario	C	E
	¿Cuántos mensajes pueden ser entendidos de manera fácil?	Porcentaje de mensajes entendidos de manera fácil sobre el total de mensajes mostrados al usuario en las implementaciones	C	E

Tabla 37. Medidas de usabilidad seleccionadas norma ISO 25023
Fuente: Elaboración propia

MANTENIBILIDAD				
Meta	Preguntas	QMEs	Tipo Métrica	Enfoque Métrica
Definir qué tan mantenible es el producto	¿Qué capacidad tiene el código de ser analizado?	Porcentaje de las funciones agregadas del total que superan la complejidad ciclomática establecida	P	I
		Porcentaje de líneas de código que poseen comentarios (del total de líneas de código lógicas)	P	I

	¿Es posible realizar pruebas sobre el software?	Porcentaje de elementos del total que poseen probabilidad del 20% de tener problemas (Según la tabla McCabe)	P	I
	¿Qué tan reutilizable es el código?	Cantidad de elementos que pueden ser reutilizados	C	I

Tabla 38. Medidas de mantenibilidad seleccionadas norma ISO 25023

Fuente: Elaboración propia

6.2. FASE DE ANÁLISIS, ADAPTACIÓN E INTEGRACIÓN DE PSP, SCRUM Y EL MODELO DE CALIDAD

6.2.1. CARACTERIZACIÓN DE SCRUM Y PSP vs SWEBOK

Para la caracterización del marco de trabajo SCRUM y el modelo PSP, se procede a desglosar las actividades y artefactos que participan en cada uno respectivamente, y se establece cual es el área de conocimiento (KA), tópico y sub-tópico correspondiente definido en el SWEBOK; a través de este mapeo de SCRUM y PSP con el SWEBOK se podrá establecer cuál es la definición dada por este para el ítem analizado, y además evaluar como SCRUM o PSP cumple con lo definido en el cuerpo de conocimientos, esto se describe en la columna modo de cumplimiento.

Algunos de los KAs y tópicos correspondientes del SWEBOK no se mapearon con SCRUM y PSP, debido a que no se encontró una relación directa con las actividades o artefactos manejados por los mismos, estos se pueden encontrar en el anexo 11.

SCRUM bajo los lineamientos del SWEBOK

La elaboración de la tabla se realizó a través de la selección del elemento a evaluar de SCRUM y con el mapeo de la respectiva área de conocimiento, tópico y sub-tópico correspondiente, a su vez se evaluó cómo se puede cumplir a través de la metodología lo definido en el SWEBOK.

ARTEFACTOS				
ÍTEM SCRUM	KA SWEBOK	Tópico SWEBOK	Sub-tópicos SWEBOK	MODO DE CUMPLIMIENTO
<ul style="list-style-type: none"> Product Backlog Sprint Backlog 	1. Requerimientos de Software	1. Fundamentos de Requerimientos de Software	1.1 Definición de un requerimiento de software 1.3 Requerimientos Funcionales y no Funcionales	La definición de SCRUM en cuanto a los requerimientos se tiene como Historias de usuario.

			1.5 Requerimientos cuantificables	La definición de las historias de usuario establece que deben ser independientes y que deben ser verificables, es decir que el cliente pueda verificar que el requerimiento o su especificación se cumplieron.
		3. Elicitación de Requerimientos	3.1 Fuentes de los requerimientos	Stakeholders, Implementadores, Desarrolladores, Propietario del Producto (roles definidos de Scrum)
			3.2 Técnicas de elicitación	-Este proceso se puede efectuar a través de Entrevistas,
		4. Análisis de requerimientos	4.1 Clasificación de Requerimientos	-Los requerimientos se pueden clasificar en : <ul style="list-style-type: none">• Mejoras• Nuevas funcionalidades• Correcciones
		5. Especificación de requerimientos	5.1 Especificación de requerimientos del software	-Los requerimientos se diligencia a través del formato de requerimientos y en la herramienta JIRA
		6. Validación de los requerimientos	6.1 Revisión de requerimientos	-Validar los requerimientos escritos con el cliente.
6.2 Prototipado	-Elaboración de prototipos como parte de la documentación de diseño para todo requerimiento			
Burndown Chart	7. Gestión de la Ingeniería de Software	3. Promulgación del proyecto de software	3.3 Implementación de los procesos de mediciones	El burndown chart es una herramienta que permite llevar un seguimiento del avance del proyecto
			3.6 Reportes	
		4. Revisión y evaluación	4.2 Revisando y evaluando el rendimiento	
ACTIVIDADES				
Sprint Planning	7. Gestión de la Ingeniería de Software	2. Planeación del proyecto de software	2.3 Esfuerzo, Calendario y Estimación del Costo	La implementación del Sprint planning es una herramienta de planeación provista por SCRUM para la asignación de recursos a las tareas, de igual forma se plantea el modelo de trabajo durante el Sprint. Aun así existen falencias de estimación en tiempo ya que SCRUM no brinda un método certero.
			2.4 Asignación de recursos	
		3. Promulgación del proyecto de software	3.3 Implementación de los procesos de mediciones	
		1. Inicio y definición del alcance	1.1 Determinación y negociación de los requerimientos	
Sprint	3. Construcción del software	2. Gestionando la construcción	2.1 Modelos de construcción	La construcción del software no tiene una definición exacta en SCRUM, solo se sabe que durante la etapa del sprint se elabora el producto planeado.
			2.3 Métricas en la construcción	
		3. Consideraciones practicas	3.2 Codificación	
Scrum Daily	7. Gestión de la Ingeniería de Software	3. Promulgación del proyecto de software	3.1 Implementación de los planes	A través del Scrum Daily se puede establecer el estado de avance del proyecto, posibles desviaciones y problemas para continuar, esto puede ir acompañado del Burndownchart
4. Revisión y evaluación		4.2 Revisando y evaluando el rendimiento		
Sprint Review		3. Promulgación del proyecto de software	3.4 Monitoreo del proceso	
		4. Revisión y evaluación	4.1 Determinando la satisfacción de los requerimientos	

Sprint Retrospective		5. Cierre	5.1 Determinando el cierre	cliente o propietario del producto, de esta manera se puede garantizar la satisfacción de los clientes.
		3. Promulgación del proyecto de software	3.5 Control del proceso	
		5. Cierre	5.2 Actividades de cierre	La retrospectiva sirve como elemento de evaluación del proceso ejecutado y posibles mejoras el cierre del sprint.

Tabla 39. SCRUM mapeado bajo los lineamientos del SWEBOK

Fuente: Elaboración propia

PSP Bajo los lineamientos del SWEBOK

La elaboración de la tabla se realizó a través de la selección del elemento a evaluar del modelo de calidad PSP en cada uno de sus niveles de acuerdo a la definición y se procedió a realizar el respectivo mapeo del área de conocimiento, tópico y sub-tópico correspondiente, a su vez se evaluó como se puede cumplir a través del modelo lo definido en el SWEBOK.

ACTIVIDADES											
Actividad Principal	Sub-actividad PSP	Nivel PSP						KA	Tópico	Sub-topicos	Interpretación
		0	0.1	1	1.1	2	2.1				
Planeación	Toma de requerimientos de programa	X	X	X	X	X	X	1. Requerimientos de software	1. Fundamentos de requerimientos de software	1.1 Definición de un requerimiento de software	En la fase de planeación el desarrollador obtiene los requerimientos asegurándose que estén claros y que no existan ambigüedades, se resuelven preguntas si es necesario, para esto el apoyo de las definiciones del SWEBOK permite establecer que se debe tener en cuenta para esto.
										1.3 Requerimientos funcionales y no funcionales	
										1.5 Requerimientos cuantificables	
									1. Inicio y definición del alcance	1.1 Determinación y negociación de los requerimientos	
									5. Especificación de requerimientos	5.3 Especificación de requerimientos de software	
	Estimación del tiempo requerido	X	X	X	X	X	X	7. Gestión de la Ingeniería de Software	2. Planeación del proyecto de software	2.3 Estimación de esfuerzo costo y calendario	Durante la fase de planeación se realizan las estimaciones de tamaño tiempo y calendario de trabajo.
	Estimación del tamaño		X	X	X	X	X				
	Uso de método PROBE			X	X	X	X				
	Planeación de tareas y calendario				X	X	X				
	Estimación de defectos					X	X				

											proyecto se realiza una proyección de los posibles defectos a encontrar, esto se cataloga como un riesgo ya que puede causar sobrecostos en la detección y la no detección.
Desarrollo	Diseño	X	X	X	X	X	X	2. Diseño de software	1. Fundamentos del diseño de software	1.1 Conceptos generales del diseño de software	La fase de planeación contiene una etapa de diseño de alto nivel, mientras que la fase de desarrollo tiene una etapa de diseño más detallado.
		1.4 Técnicas requeridas	PSP 2.1 guía los diseños a través de plantillas especificadas.								
		2. Aspectos importantes en el diseño de software	2.1 Concurrencia	Es importante tener en cuenta estos elementos para el desarrollo del diseño, PSP no los especifica pero es importante tenerlos en cuenta para el desarrollo del diseño.							
			2.2 Control y manejo de eventos								
			2.3 Distribución de componentes								
			2.4 Manejo de errores y tolerancia a fallos								
			2.5 Interacción y presentación								
			2.6 Persistencia de datos								
		5. Notaciones del diseño de software	5.1 Vistas estáticas								
			5.2 Vistas dinámicas								
		3. Construcción del software	2. Gestionando la construcción	2.2 Planeando la construcción	Esto se desarrolla en las fases de planeación y diseño de PSP						
			3. Consideraciones prácticas	3.1 Diseño en la construcción							
	Codificación	X	X	X	X	X	X		1. Fundamentos de la construcción de software	1.1 Minimizando la complejidad	Los estándares de codificación son una herramienta que ayude a cumplir con estos objetivos.
										1.2 Anticipando al cambio	
										1.3 Construyendo para la verificación	
										1.4 Estándares en la construcción	
									2. Gestionando la construcción	2.1 Modelos de construcción	Dentro de la fase de desarrollo de PSP se llevan métricas de defectos y tiempos de desarrollo.
										2.3 Métricas en la construcción	
3.									3.2 Lenguajes de	En PSP es	

									Consideraciones prácticas	construcción	necesario definir el lenguaje de construcción de los programas que permitan unificar las métricas de estimación de tamaño.
										3.2 Codificación	Etapas de desarrollo
										3.4 Reuso	Debe estar definido dentro de los estándares de codificación
	Compilación	X	X	X	X	X	X				
	Pruebas	X	X	X	X	X	X	4. Pruebas de software	1. Fundamentos de las pruebas de software	1.1 Terminología relacionada 1.2 Aspectos claves	PSP define en la fase de desarrollo la etapa de pruebas unitarias, las cuales son planificadas y ejecutadas por cada el desarrollador de la tarea.
								2. Niveles de pruebas	2.1 El destino de las pruebas 2.1.1 Pruebas unitarias 2.2 Objetivos de las pruebas 2.2.1 Pruebas de aceptación 2.2.4 Pruebas de conformidad, pruebas funcionales		
								3. Técnicas para las pruebas	3.1 Basado en la experiencia del ingeniero de software		
								5. Proceso de las pruebas	5.1 Consideraciones prácticas 5.1.2 Guías para las pruebas 5.1.4 Documentación de las pruebas y productos de trabajo 5.1.7 Terminación de las pruebas		
										5.2 Actividades de las pruebas	
	Registro de tiempos en fases	X	X	X	X	X	X	3. Construcción del software	2. Gestionando la construcción	2.3 Métricas en la construcción	Dentro de la fase de desarrollo de PSP se llevan métricas de defectos y tiempos de desarrollo.
	Registro de defectos en fases	X	X	X	X	X	X				
	Revisiones de diseño					X	X	2. Diseño de software	4. Análisis y evaluación del	4.2 Análisis de la calidad y	

									diseño de software	técnicas de la evaluación	
	Revisiones de código					X	X	3. Construcción del software	1. Fundamentos de la construcción de software	1.3 Construyendo para la verificación	Las revisiones de código son un elemento fundamental en PSP con que permite la detección temprana de errores y además permite verificar el cumplimiento de los estándares de codificación
Postmortem	Revisión de registro de Defectos	X	X	X	X	X	X	7. Gestión de la ingeniería de software	3. Promulgación del proyecto de software	3.1 Implementación de los planes	La fase de postmortem sirve para verificar la calidad de los datos obtenidos
	Revisión de consistencia de datos en defectos	X	X	X	X	X	X			3.3 Implementación del plan de métricas	
	Revisión de registro de tiempos	X	X	X	X	X	X				
	Recolección de tamaño del programa		X	X	X	X	X				
ARTEFACTOS											
	Resumen de plan del proyecto	X	X	X	X	X	X	7. Gestión de la ingeniería de software	2. Planeación del proyecto de software	2.3 Estimación de esfuerzo, calendario y costos	La platilla de resumen de proyecto permite tener de manera completa el panorama de estimación del proyecto.
	Log de registro de tiempos	X	X	X	X	X	X		3. Promulgación del proyecto de software	3.6 Reportes	Los logs de registros de tiempos y defectos hacen parte fundamental de la construcción de reportes.
	Log de registro de defectos	X	X	X	X	X	X				
	Estándar de tipos de defectos	X	X	X	X	X	X				
	Estándar de codificación		X	X	X	X	X	3. Construcción del software	1. Fundamentos de la construcción de software	1.4 Estándares en la construcción	PSP y SWEBOK definen que es importante tener un estándar de codificación a seguir que permita a los desarrolladores ceñirse a un modelo y normas de codificación.
	PIP (Problemas y mejoras en el proceso)		X	X	X	X	X	10. Calidad de Software	1. Fundamentos de la calidad de software	1. Mejora de la calidad	El PIP es una herramienta fundamental que permite al desarrollador evaluar su progreso y plantear propuestas de mejora personales.
	Plantilla de			X	X	X	X	7. Gestión de la	2. Planeación	2.3 Estimación	Se puede definir

	estimación de tamaño							ingeniería de software	del proyecto de software	de esfuerzo, calendario y costos	que la plantilla de estimación de tamaño es el corazón es la base que permitirá estimar los otros elementos como tiempo y defectos.
	Plantilla de Reporte de Pruebas			X	X	X	X	4. Pruebas de Software	5. Proceso de las pruebas	5.1 Consideraciones practicas 5.1.4 Documentación de las pruebas y productos de trabajo	Uno de los mecanismos que permite llevar unas pruebas unitarias ordenadas es la plantilla de reporte de pruebas, además permite llevar un registro de las pruebas realizadas que permitan ser guía para futuras reproducciones como lo especifica el SWEBOK.
	Plantilla de tareas				X	X	X	7. Gestión de la ingeniería de software	2. Planeación del proyecto de software	2.3 Estimación de esfuerzo, calendario y costos	Otro mecanismo que ayuda a la organización en PSP es la plantilla de calendario y tareas, de tal manera que se pueda llevar un control del trabajo y su avance.
	Plantilla de Calendario				X	X	X				
	Lista de chequeo de revisiones de diseño					X	X	2. Diseño de software	4. Analisis y evaluación del diseño de software	4.2 Análisis de la calidad y técnicas de la evaluación	Las listas de chequeo son vitales para llevar un proceso ordenado y secuencial para la verificación de los productos elaborados.
	Lista de chequeo de revisiones de codigo					X	X	3. Construcción del software	1. Fundamentos de la construcción de software	1.3 Construyendo para la verificación	
	Plantillas de diseño <ul style="list-style-type: none"> Plantilla de especificación operacional Plantilla de especificación funcional Plantilla de especificación de estados Plantilla de especificación lógica 						X	2. Diseño de software	5. Notaciones del diseño de software	5.1 Vistas estáticas	PSP define una serie de plantillas como guías de diseño para los desarrolladores.
										5.2 Vistas dinámicas	
	Estándar de defectos						X	10. Calidad de Software	3. Consideraciones	3.2 Caracterización	PSP caracteriza los defectos de tal

	extendido									Practicas	de defectos	manera que puedan ser recopilados de manera estandarizada en las etapas de desarrollo.
--	-----------	--	--	--	--	--	--	--	--	-----------	-------------	--

Tabla 40. PSP mapeado bajo los lineamientos del SWEBOK

Fuente: Elaboración propia

De acuerdo al mapeo realizado de SCRUM y PSP, frente al SWEBOK, se realizó el siguiente grafico que permite visualizar los porcentajes de cubrimiento de cada uno frente al cuerpo de conocimientos de la ingeniería de software, se puede observar como PSP encuentra varios elementos mapeados en el SWEBOK correspondientes a la fase de construcción del producto de software mientras que SCRUM prevalece en los elementos de requisitos e ingeniería de software, lo cual fue base para establecer los puntos fuertes de cada modelo para su integración; por otro lado también se puede visualizar que faltaron aun muchos elementos por mapear, los cuales no se les encontró una relación directa con las actividades y artefactos que se mapearon tanto de SCRUM como de PSP (Anexo 11), debido a que en esta primera etapa del proyecto no se entro en un detalle más profundo de estas áreas del conocimiento y también su relevancia no se estableció como considerable para el proyecto.

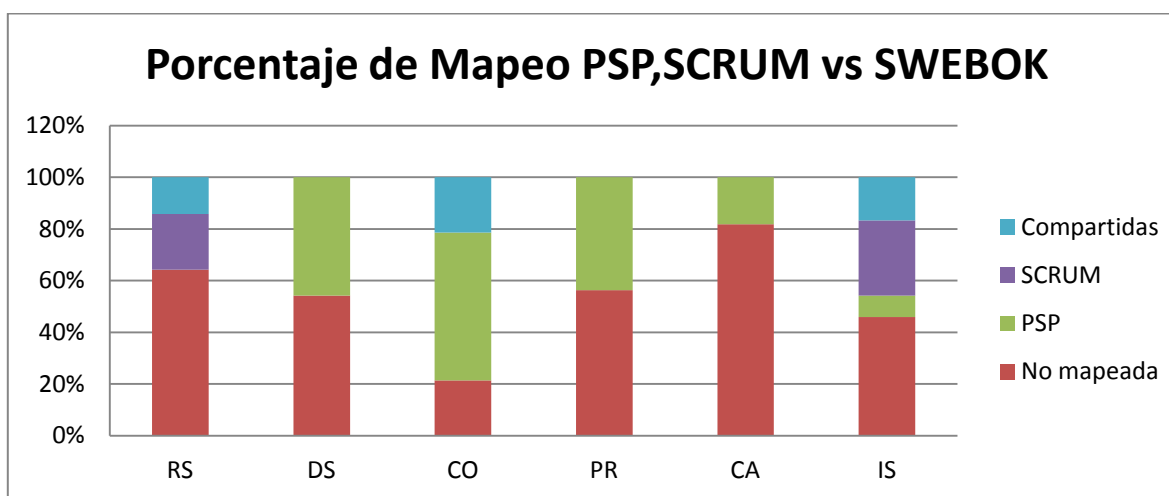


Figura 38. Porcentaje de Mapeo de PSP y SCRUM vs SWEBOK, Fuente: Elaboración propia

Por otro lado de acuerdo a los cuadros anteriores, se elaboraron los siguientes cuadros gráficos, que permiten visualizar el mapeo de definición dado en SWEBOK con PSP y SCRUM, de acuerdo a la siguiente carta de colores se establece si un área del conocimiento, tópico y sub-tópico define alguna actividad o artefacto de los modelos analizados.

Color	SCRUM	PSP
	N/A	N/A
	X	
		X
	X	X

Tabla 41. Carta de colores de mapeo SCRUM y PSP frente al SWEBOK

ÁREAS DEL SWEBOK

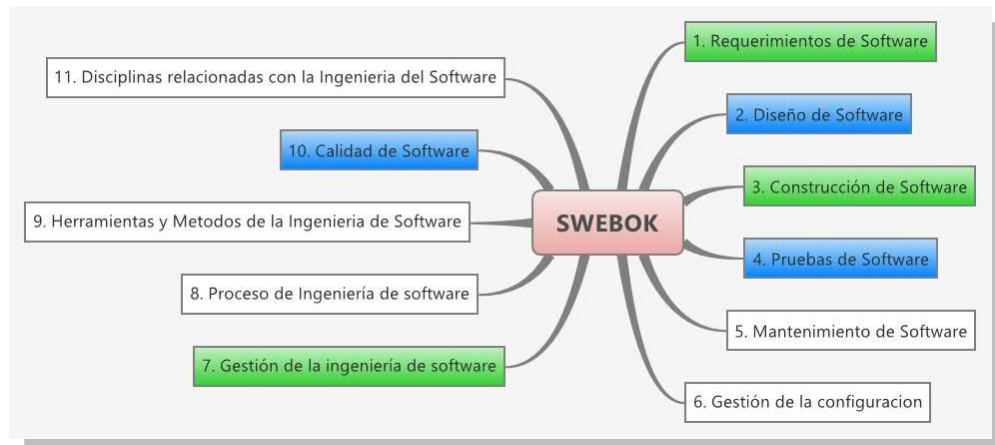


Figura 39. Vista General de las Áreas del Conocimiento del SWEBOK

De la anterior gráfica se puede observar que SCRUM y PSP tienen definiciones en común encontradas en el SWEBOK (Resaltadas en verde), y hay algunas que solo son definidas por PSP, de lo cual se puede confirmar que PSP es un elemento que puede complementar a SCRUM en las falencias que pueda presentar.

TÓPICOS Y SUBTÓPICOS DE CADA ÁREA DEL SWEBOK

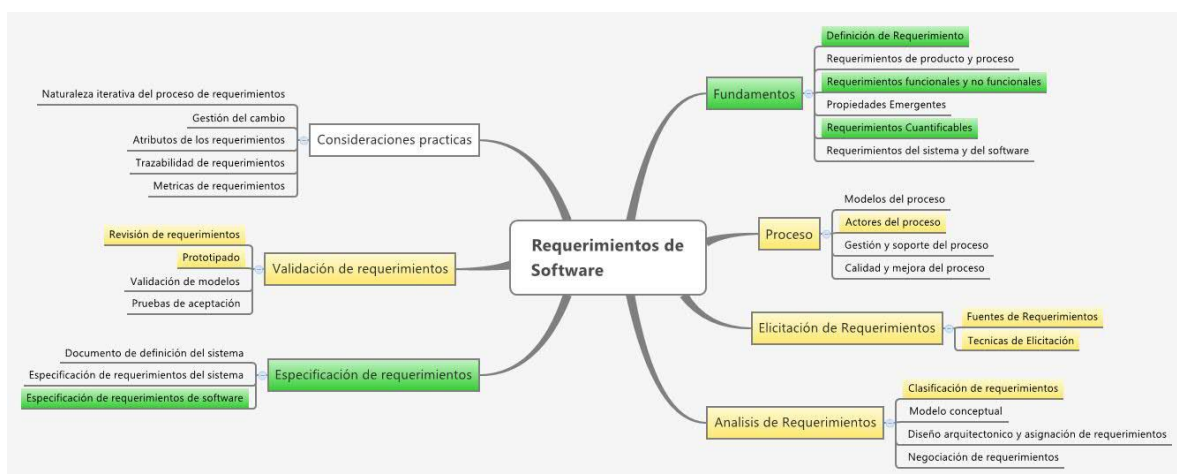


Figura 40. Área del conocimiento requerimientos de software

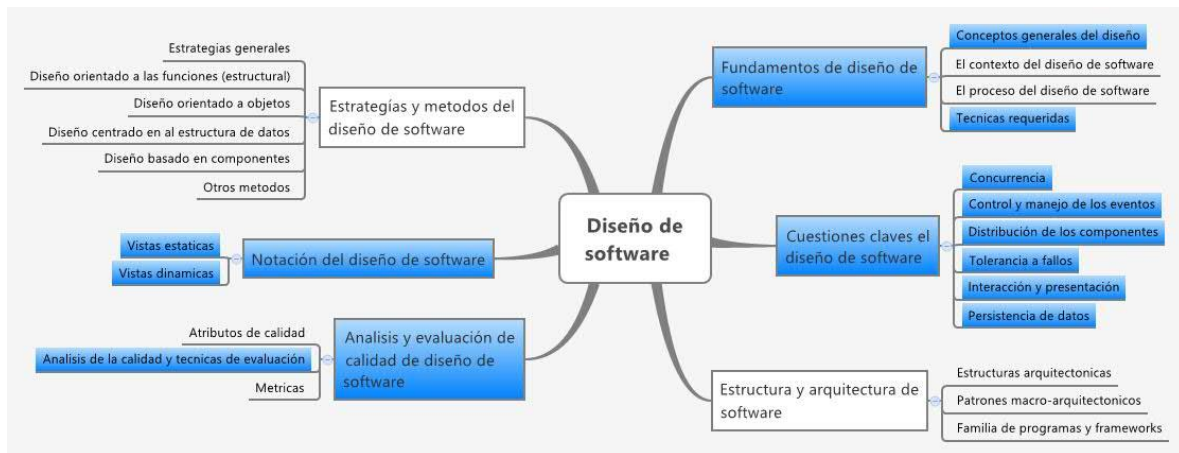


Figura 41. Área del conocimiento diseño de software



Figura 42. Área del conocimiento construcción del software

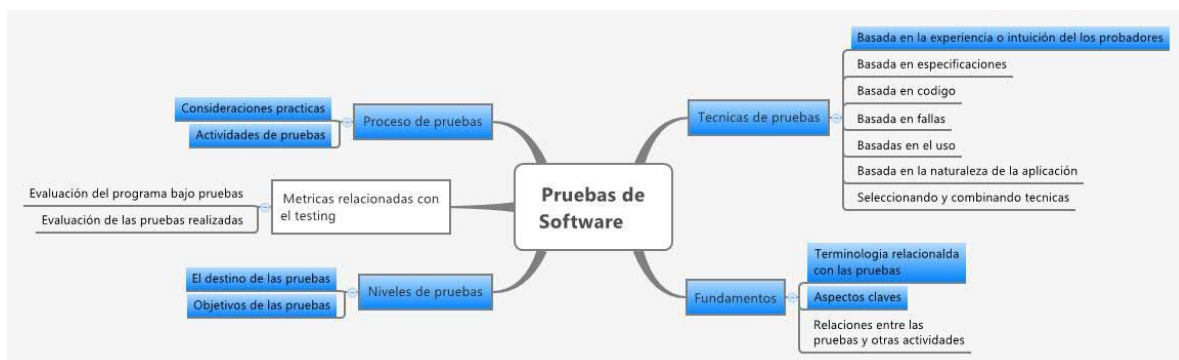


Figura 43. Área del conocimiento pruebas de software

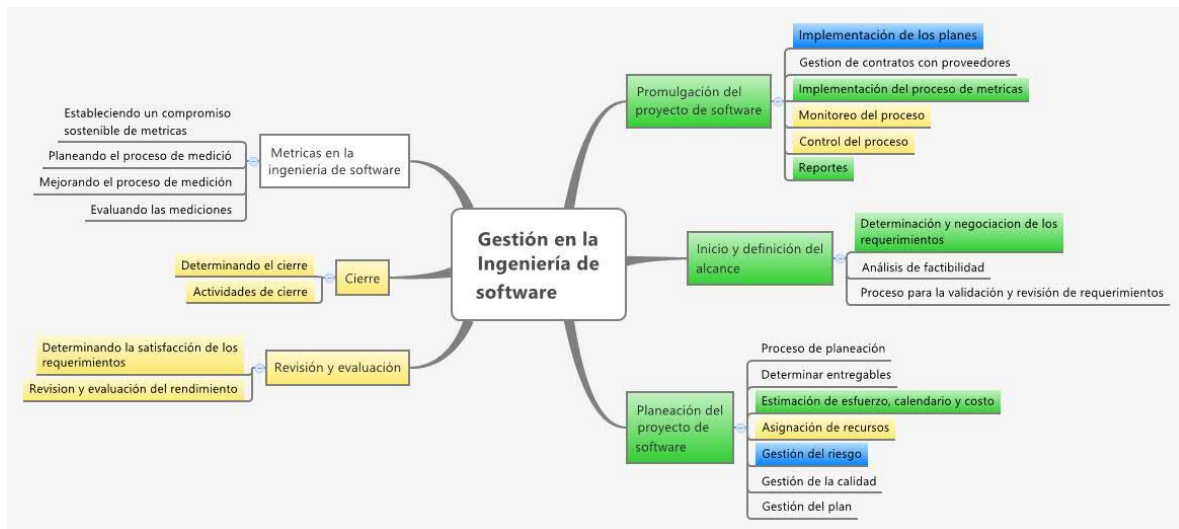


Figura 44. Área del conocimiento gestión de la ingeniería de software

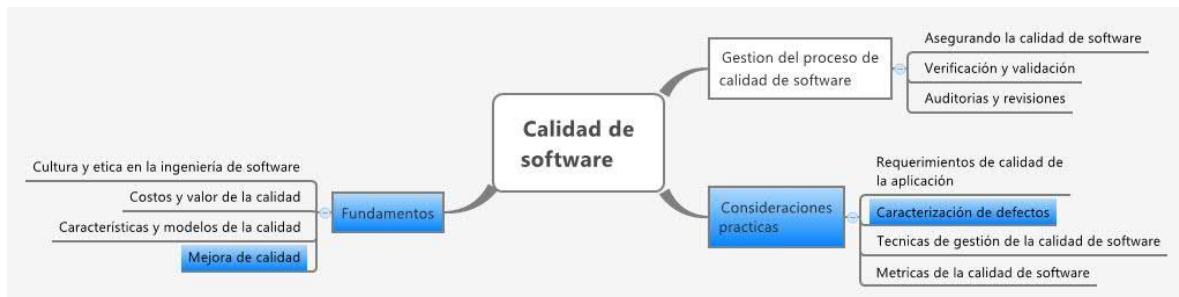


Figura 45. Área del conocimiento calidad del software

De acuerdo a las anteriores gráficas se puede resaltar que las áreas comunes compartidas entre SCRUM y PSP están concentradas en el área de requerimientos, construcción e ingeniería del software, mientras PSP sobresale en las áreas de diseño, pruebas y calidad, reafirmando lo dicho anteriormente de que PSP es un elemento que puede complementar de manera integral a SCRUM; Por otro lado se puede concluir que los elementos presentados en los dos modelos tienen referencias teóricas validas basadas en el SWEBOK, de tal manera que pueden ser complementadas más aun las definiciones de cada uno.

6.2.2. MAPEO DE SCRUM Y PSP

A continuación se realiza una tabla donde se procede a evaluar las actividades propuestas en el marco de trabajo SCRUM frente a PSP2.1 (se toma este nivel ya que posee todas las actividades previas condensadas), de tal manera que permita evidenciar si existen actividades que compitan o complementen entre sí de acuerdo a la fase del proceso que correspondan (planeación, desarrollo y *postmortem*):

RELACIÓN	DESCRIPCIÓN
E	Actividad PSP que tiene similitud con relación a la comparada en SCRUM y puede llegar a reemplazar o ser reemplazada
C	Actividad PSP que no tiene similitud sino que puede definirse como complemento de la actividad SCRUM
N	No definida

Tabla 42. Tipo de relaciones de actividades, roles y artefactos de SCRUM y PSP

Área SWEBOK / Topico	Actividad SCRUM	Actividad PSP	Tipo Relación	Actividad Seleccionada		Observaciones
				SCRUM	PSP	
Gestión en la ingeniería del software / Planeación del proyecto de software	<i>Sprint Planning</i> (Presentación <i>product backlog</i>)	Toma de requerimientos	E	X		Se selecciona la actividad de SCRUM ya que permite al equipo conocer y realizar aportes al trabajo asignado a cada desarrollador durante el desarrollo del Sprint
	<i>Sprint Planning</i> (Estimación por <i>Planning Pocker</i>)	Estimación de tamaño (LOC)	C	X	X	La actividad de <i>planning pocker</i> (por tallas) es una actividad en el cual se puede estimar el tamaño de una tarea a desarrollar en el Sprint de manera global y rápida a través con el consenso de todo el equipo, basada en la experiencia, y puede complementarse haciendo uso de la estimación de tamaño a través del método PROBE para detallar esta estimación de tamaño, por otro lado es requerida para la estimación de tiempo a través de este mismo método.
	<i>Sprint Planning</i> (Estimación de tiempo)	Estimación de tiempo requerido (PROBE)	E		X	La estimación de tiempo requerido a través del método PROBE de PSP es una herramienta que ha demostrado ser muy efectiva y acercada a la realidad, de acuerdo a los resultados presentados por el SEI
	<i>Sprint Planning</i> (Establecer tareas)	Establecer tareas y planeación del calendario	E	X		En la reunión de <i>Sprint planning</i> se puede tener un apoyo de todo el equipo para dividir cada tarea en sub-tareas, de tal manera que se permita tener varios puntos de vista por parte de los desarrolladores, establecer el calendario puede convertirse en una actividad extra que puede volver pesado el proceso, además Sprint posee unas fechas ya definidas de entrega y a través del <i>Scrum Daily</i> se puede evaluar el avance del proyecto.
	-	Estimación de defectos	N		X	Se usara el método de estimación de defectos de PSP ya que es una herramienta que permitirá ayudar al proceso de estimación de tiempos, riesgos y costos de calidad.
Gestión de la ingeniería de software / Revisión y evaluación	<i>Scrum Daily</i>	Diligenciar la plantilla de calendario desarrollada	E	X		La actividad del <i>Scrum Daily</i> es una actividad permite realizar un monitoreo de manera diaria al desarrollo del sprint por todo el equipo de tal manera que todos están informados del avance del proyecto, por otro lado permite que se puedan corregir posibles problemas o desviaciones en lo planeado.
Construcción del Software / Gestionando la construcción	<i>Sprint</i>	Desarrollo <ul style="list-style-type: none"> Diseño 	C	X	X	Permite que el desarrollador haga un diseño conceptual a una más baja escala de los requerimientos para que guíen su codificación de una manera más acertada, se utilizara un plantilla básica de diseño

						que permita guiar al desarrollador en la elaboración de un diseño detallado y además que permita llevar una documentación que sea visible en el proyecto.
Diseño de Software / Análisis y evaluación del diseño de software		Revisión de diseño	C			No se selecciona esta actividad con el fin de no volver demasiado pesado el proceso de desarrollo por otro lado ya existe un diseño previo realizado a través de prototipos y casos de uso que entran como material del product backlog según lo definido en el proceso, además no es una actividad crítica en el proceso.
Construcción del Software / Fundamentos de la construcción del software		Codificación	E	X	X	Esta actividad está emulada de manera inherente en los dos modelos ya que es la que permite construir el producto final.
		Revisiones de código	C		X	Se incluye esta actividad para que cada desarrollador evalúe el cumplimiento de los estándares de codificación por parte suya, y pueda detectar de manera temprana posibles defectos.
-		Compilación	C		X	Esta actividad se incluye como parte del proceso de desarrollo ya que actualmente el IDE requiere una compilación del proyecto para encontrar posibles errores, (Esto se omitirá cuando se trabaje en IDEs más avanzados como Visual Studio .NET sea extenso.
Pruebas de Software / Proceso de las pruebas - Fundamentos de pruebas de software		Pruebas	C		X	Esta actividad es una de las más importantes ya que permite al desarrollador realizar pruebas unitarias sobre el trabajo realizado y detectar a tiempo posibles defectos para corregirlos antes de liberar el producto.
Construcción del Software / Gestionando la construcción	<i>Sprint</i> (Registro de tiempo restante en fases)	Registro de tiempos en fases	E	X	X	Esta actividad es clave para evaluar el avance del proyecto a través del <i>Burndown chart</i> de SCRUM y además para registrar datos históricos de los tiempos de desarrollo, para futuras estimaciones.
		Registro de defectos en fases	N		X	Esta actividad permite al desarrollador llevar un registro de cuáles son los errores más comunes que comente durante cada una de las fases, y de esta manera establecer acciones correctivas que ayuden a incrementar su nivel de calidad.
Gestión de la ingeniería de software / Promulgación del proyecto	-	Revisión de registro de defectos	N		X	A través de esta actividad permitirá al desarrollador evaluar la calidad de los registros de defectos realizados y además enterarse de los errores cometidos en cada una de las fases y donde se corrigieron.
	-	Revisión de consistencia de datos de defectos	N		X	A pesar que se use la herramienta PSP, se hace necesario realizar revisiones en la consistencia de los tiempos ya que a veces existen problemas de registros incompletos por caídas en el sistema o la energía..
Construcción del Software / Gestionando la construcción	-	Recolección de tamaño	N		X	La recolección del tamaño del código realizado, permitirá obtener métricas más acertadas para la estimación de los tiempos, y de esta manera utilizar esta información junto a los resultados de la clasificación por tamaños dada por el <i>planning poker</i> .
Gestión de la ingeniería de software / Promulgación del proyecto	-	Revisión de registro de tiempos	C		X	A pesar que se use la herramienta PSP, se hace necesario realizar revisiones en la consistencia de los tiempos ya que a veces existen problemas de registros incompletos por caídas en el sistema o la energía.
*Gestión de la	<i>Sprint Review</i>	-	N	X		Esta actividad permite mostrar al

ingeniería del software / Cierre *Calidad de software / Fundamentos de la calidad del software						propietario del producto los resultados de los requerimientos inicialmente planteados en el <i>Sprint Planning</i> , permitiendo que haya una entrega formal con la respectiva retroalimentación
	Retrospectiva del <i>Sprint</i>	Postmortem (Completar el PIP)	C	X	X	Esta actividad permite al equipo evaluarse a nivel de equipo y plantear cuales problemas se presentaron, con sus respectivas propuestas de mejora para el siguiente <i>Sprint</i> , por otro lado la fase de <i>postmortem</i> de PSP define el PIP como la herramienta que permite al desarrollador hacer una retrospectiva de menor escala, para que el evalúe su proceso y plantee posibles ideas de mejoramiento, que puede ser complementario para la retrospectiva del sprint.

Tabla 43. Actividades seleccionadas de SCRUM y PSP para integración SCRUM-P

6.2.3. INTEGRACIÓN DE SCRUM Y PSP

De acuerdo a la tabla anterior en la cual se evaluaron las actividades de SCRUM y PSP, se lograron establecer las actividades principales que se van a utilizar dentro de la integración del proceso, para lo cual se elaboró el siguiente diagrama de procesos donde se puede observar de manera más clara como queda integrado el proceso.

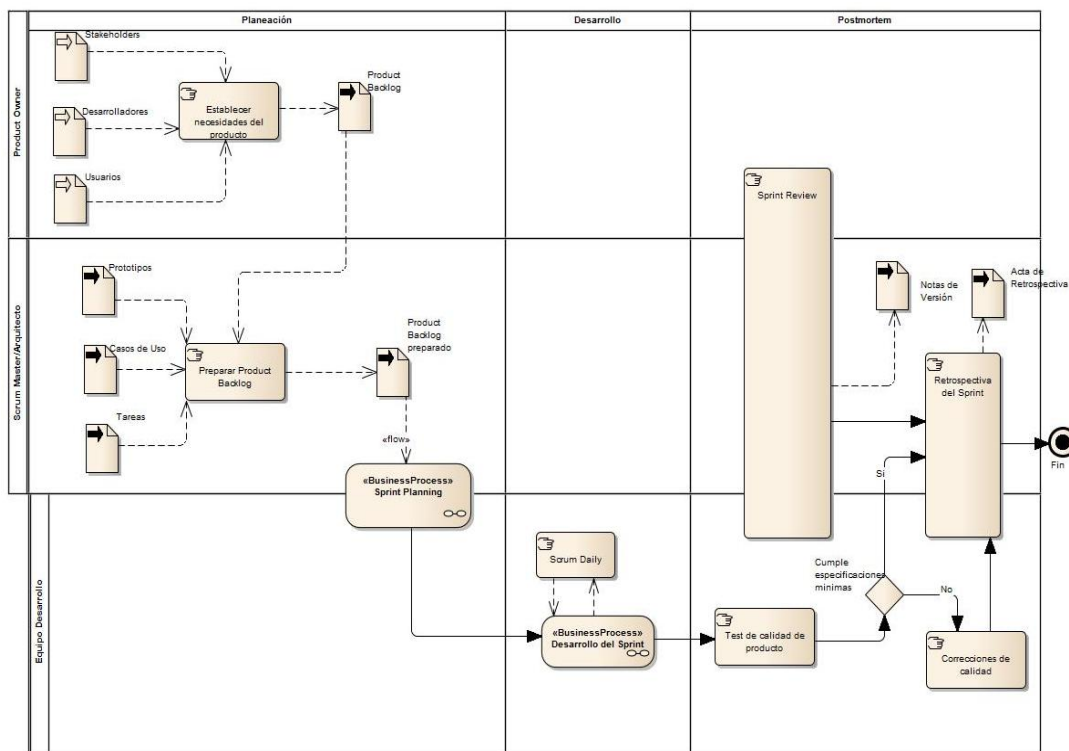


Figura 46. Diagrama de proceso general SCRUM-P

En el gráfico anterior se muestra a nivel macro el proceso integrado de SCRUM y PSP, este está compuesto desde las actividades donde se toman los requerimientos hasta la salida del producto final, a continuación se explican cada una de las actividades realizadas:

- **Establecer necesidades del producto:** En esta actividad es donde entran los requerimientos del usuario para iniciar a hacer parte del *product backlog* (sin documentación) de la organización, estos son obtenidos de diversas fuentes como *stakeholders*, implementadores, desarrolladores, *product owner*.
- **Preparación del *product Backlog*:** Una vez se tenga el *product backlog* se procede a prepararlo y priorizarlo de acuerdo al criterio dado por el propietario del producto según los lineamientos de SCRUM, con el fin de complementarlo con documentación a través de casos de uso, prototipos y sub-tareas.

Sprint Planning: En esta actividad se procede a realizar el proceso de planeación del Sprint que va a darle un incremento al producto, esta actividad se visualiza a nivel detallado en la figura 45, y se explica a continuación:

- **Establecer tiempos del equipo:** Se procede a establecer las fechas de inicio y finalización del *Sprint*, de igual forma la disponibilidad de tiempo de los miembros del equipo, con el fin de lograr establecer el total de horas-hombre de trabajo disponibles para realizar la planeación del *Sprint*.
- **Selección de tareas y explicación:** Se procede a explicar las tareas que se van a incluir en el Sprint, se resuelven dudas y se presenta la documentación anexa (Prototipos, casos de uso, diagramas de bases de datos).
- **Asignación de tareas:** Se procede a asignar las respectivas tareas a los desarrolladores de acuerdo a la disponibilidad.
- **Planeación PSP:** Cada desarrollador de acuerdo a la explicación que se le dio de cada tarea, la documentación entregada, procede a realizar el proceso de planeación PSP (Figura 46) modificado con el fin de lograr estimar el esfuerzo en tiempo para el desarrollo de las tareas.
- **Finalizar el Sprint planning:** Una vez todos los desarrolladores tengan las estimaciones de sus tareas, se evalúa si quedaron tareas por fuera por falta de recursos de tiempo y se toma acción, se da inicio al sprint.

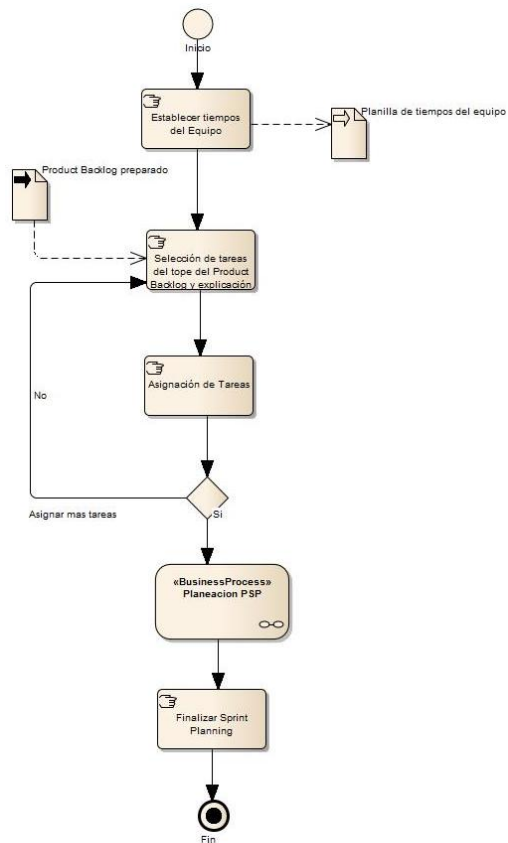


Figura 47. Diagrama de proceso *sprint planning* SCRUM-P Fuente: Elaboración Propia

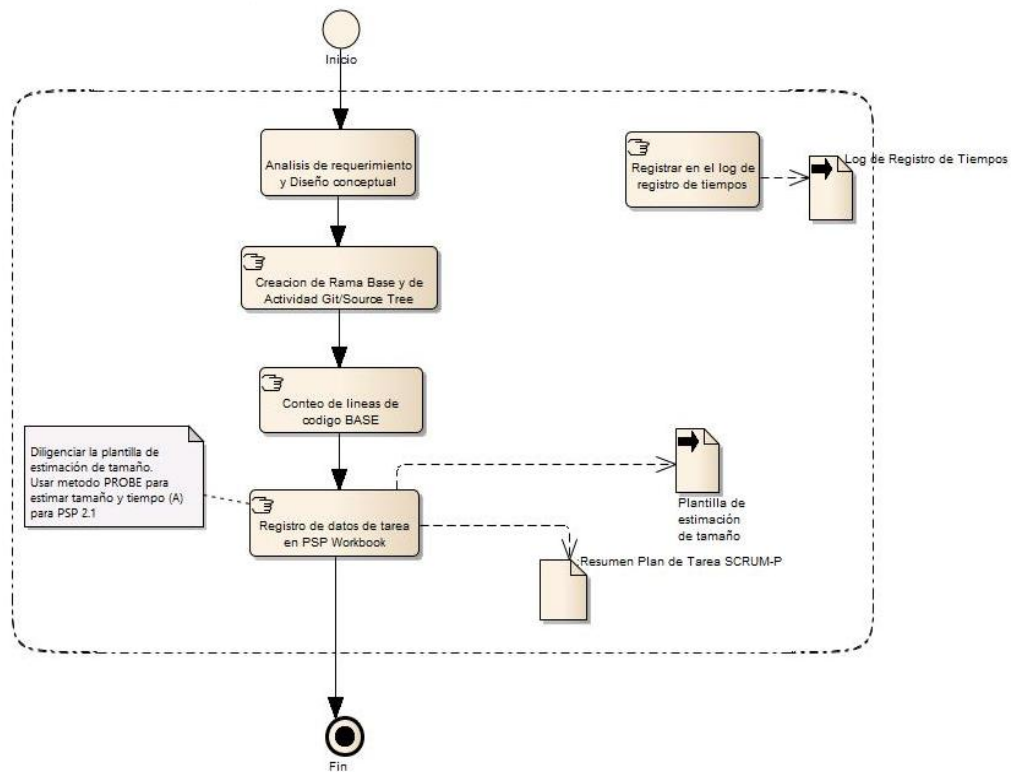


Figura 48. Diagrama de proceso Planeación PSP

- **Desarrollo del *sprint*:** En esta actividad en donde se procede a construir el incremento del producto planeado en el *Sprint planning*, como se muestra en la figura 47 se constituye de varias partes las cuales se describen a continuación:
 - **Seleccionar la tarea a desarrollar:** En esta tarea cada desarrollador selecciona una de las tareas asignadas durante el *Sprint planning*, de acuerdo a la prioridad del *Sprint Backlog* y como entrada tiene la documentación elaborada para el mismo.
 - **Desarrollo de la tarea:** El desarrollo de la tarea es el proceso en el cual los desarrolladores construyen cada una de los incrementos del producto planeados, este proceso se describe más detalladamente de manera visual en la figura 48, donde se aplican las actividades de PSP seleccionadas de la fase de desarrollo para la integración con SCRUM.
 - **Postmortem:** Esta tarea corresponde a las actividades finales del desarrollo de cada tarea y allí es donde se encuentra la aplicación de los elementos de la fase de PSP que lleva su mismo nombre, además de la evaluación de calidad del producto. Esta tarea se describe a nivel más detallado en la figura 50.

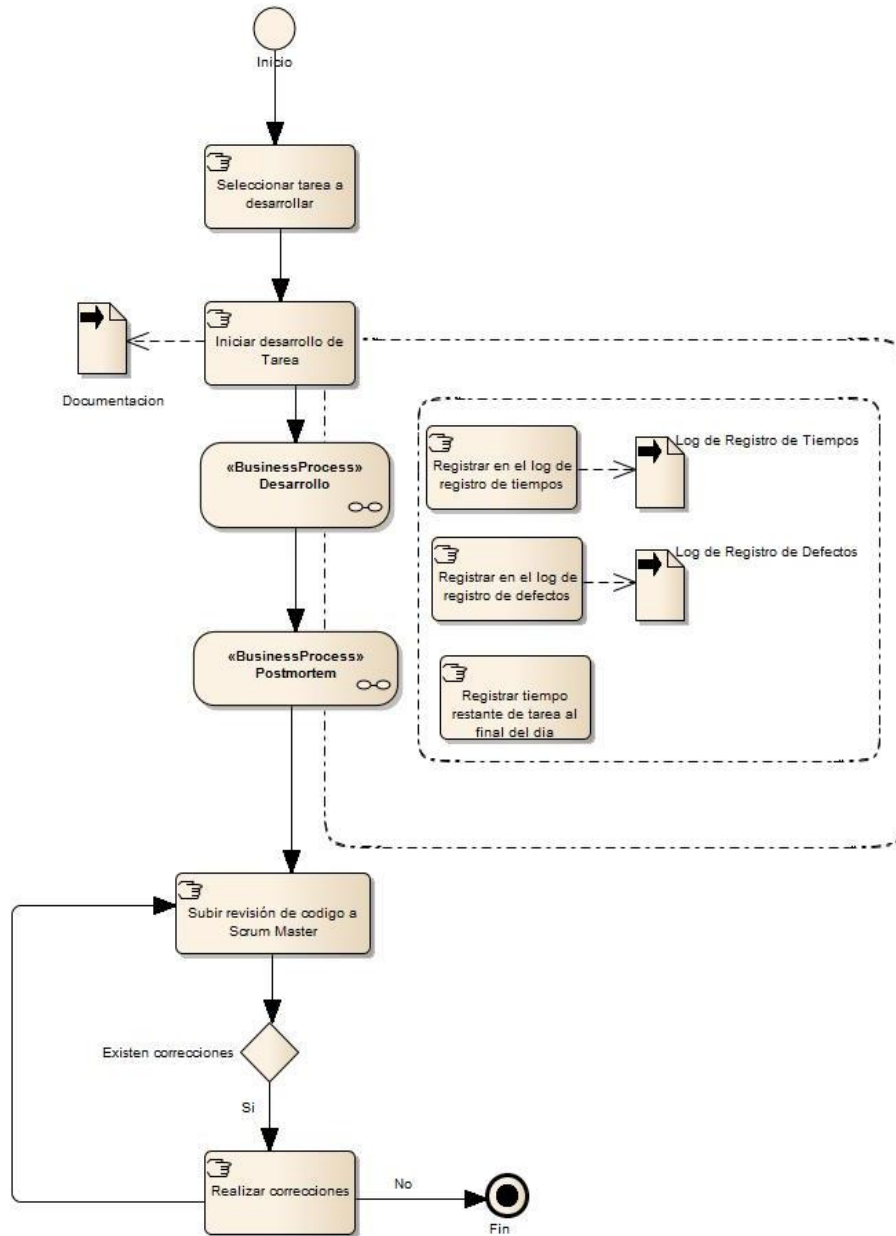


Figura 49. Diagrama de proceso de desarrollo de tareas del sprint

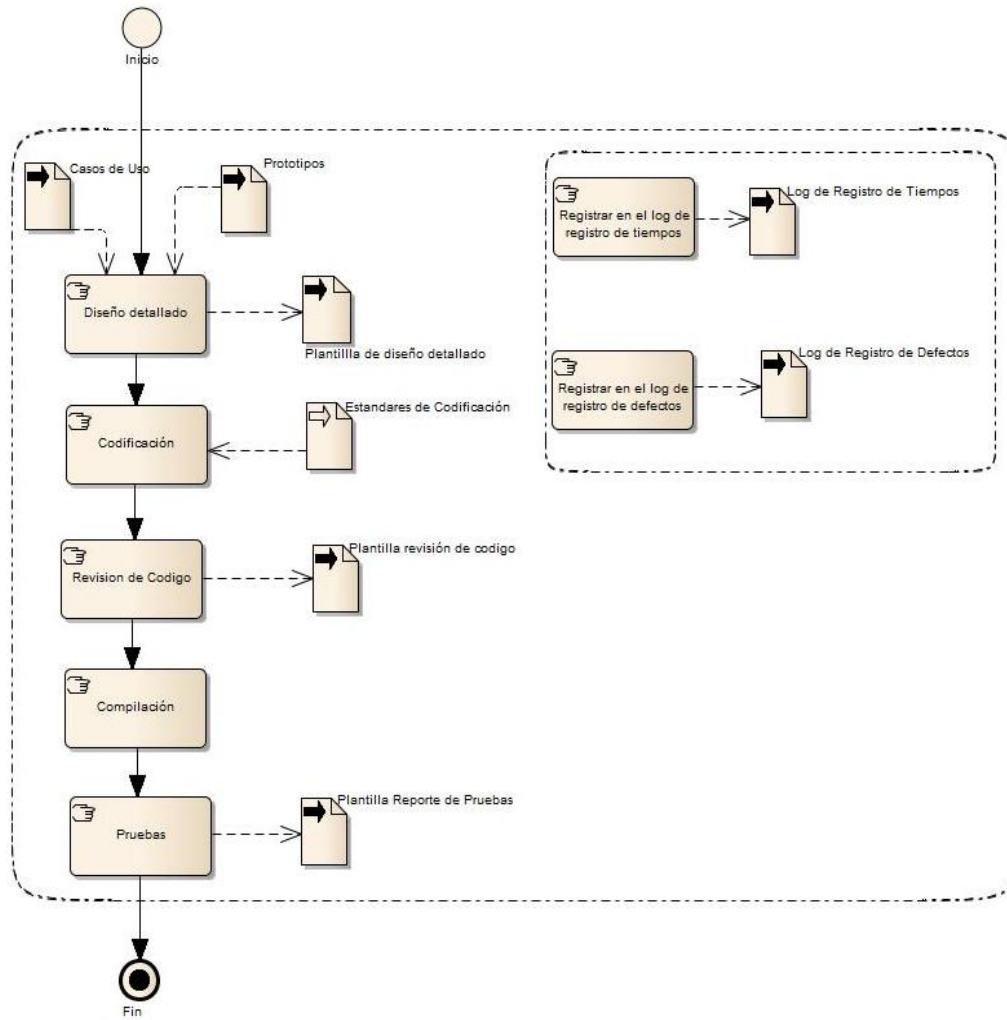


Figura 50. Diagrama de proceso desarrollo detallado de tarea

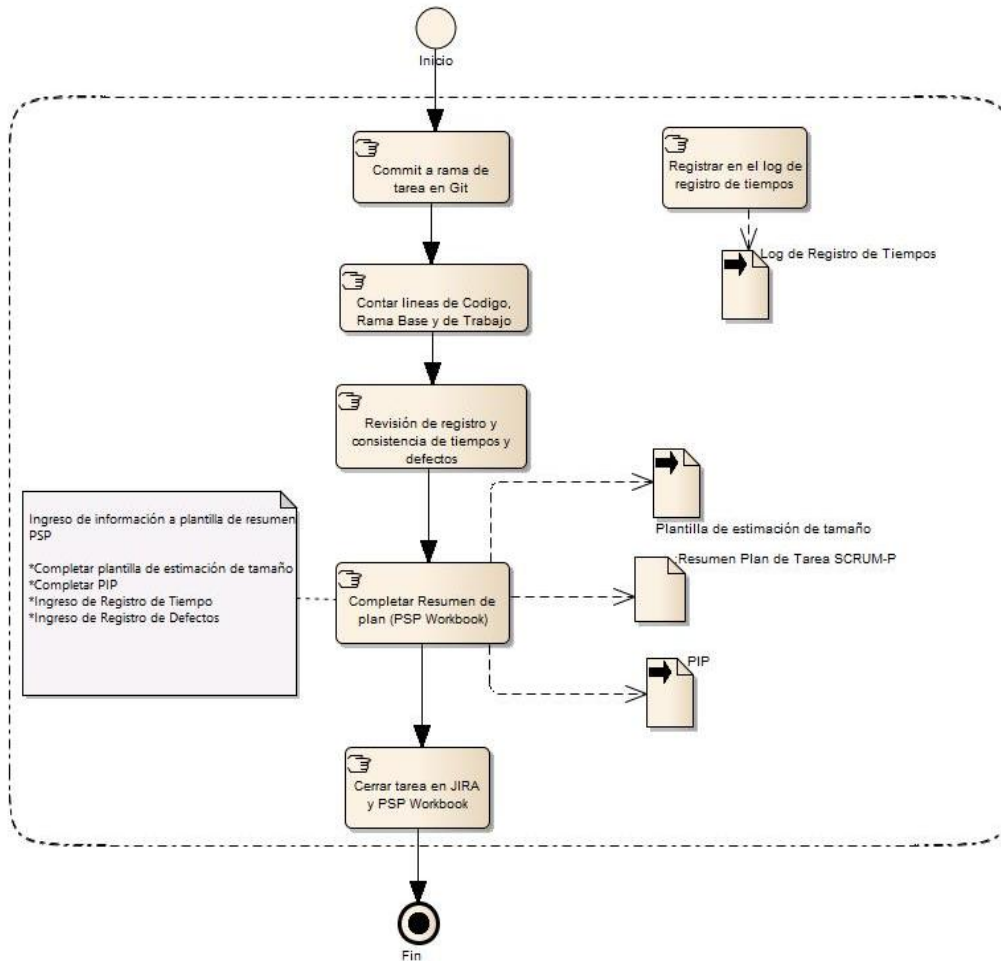


Figura 51. Diagrama de proceso postmortem de tarea

- **Test de calidad del producto:** Esta actividad correspondiente al proceso principal (Figura 45) corresponde a la etapa en donde se realizan las pruebas de calidad según los criterios seleccionados al incremento del producto – de acuerdo con el modelo ISO. Si no se cumplen se deben realizar las correcciones de calidad correspondientes para seguir a la siguiente fase, para esta actividad se puede consultar el anexo 12 en el cual se especifican las herramientas utilizadas para este proceso y la descripción de su respectivo uso.
- ***Sprint Review:*** Corresponde a la presentación del incremento del producto al propietario del producto.
- ***Retrospectiva del Sprint:*** Es la actividad donde el equipo evalúa como estuvo el Sprint y que se debe mejorar, también se evalúa como estuvo el proceso de PSP y que se puede mejorar del proceso de acuerdo a la experiencia de los desarrolladores durante el *Sprint*.

6.3. RESULTADOS FASE PILOTO

Los resultados de la fase piloto se presentaran a continuación en donde se detallan los resultados de la implementación del proceso SCRUM-P identificando las mejoras aportadas por cada uno de los procesos, de igual forma se obtuvieron resultados de datos importantes sobre las ventajas, desventajas y dificultades de implementación, que permitirán a otras empresas utilizar esta información para mejorar sus procesos de desarrollo y control de la calidad de software.

La implementación del proceso SCRUM-P se realizó a través del desarrollo de un *Sprint* en la organización, donde se aplicaron y usaron las actividades definidas previamente en el proceso, a continuación se mencionan algunas particularidades para la implementación del proceso:

- Introducción del proceso al equipo: La inducción del equipo a la metodología se realizó a través de la explicación individual de SCRUM y PSP por separado estableciendo las características de cada uno, ventajas y actividades realizadas, posteriormente se explica cómo se integraron las actividades que se seleccionaron de cada uno, y como se va a realizar la implementación del proceso integrado a través del desarrollo del próximo *Sprint* en la organización
- Características del *Sprint*:
 - Duración: 21 Días
 - Proyecto: Mejoras y Nuevas funcionalidades aplicación principal GBM
 - Miembros del Equipo: 4 Desarrolladores
 - Número de Tareas: 23 Tareas

6.3.1. HERRAMIENTAS UTILIZADAS

Como se mencionó anteriormente la organización cuenta con ciertas herramientas que permitirán facilitar el proceso de implementación de la integración de SCRUM y PSP, en donde se modificaron y añadieron algunas herramientas de acuerdo a los procesos definidos anteriormente, estas herramientas se mencionan a continuación:

Nombre de Herramienta	Añadida (S/N)	Modificada (S/N)	Descripción de modificación	Funcionalidad final
JIRA Provista Atlassian Figura 50	N	N		<ul style="list-style-type: none">• Gestión del <i>product backlog</i>.• Desarrollo del <i>Sprint planning</i>.• Actualización de desarrollo de tareas

				<p>a través del <i>Kanban Board</i>.</p> <ul style="list-style-type: none"> Reportes de <i>Burndown Chart</i> y finalización del <i>Sprint</i>.
FishEye/Crucible Provista Atlassian Figura 51	N	N		<ul style="list-style-type: none"> Revisiones de código individuales y final del <i>Scrum Master</i>.
Git	N	N		<ul style="list-style-type: none"> Herramienta de repositorio y control de código
PSP (Aplicación propia)	N	S	<p>Se ajusta la herramienta para facilitar:</p> <ul style="list-style-type: none"> Cambios de fases en registro de tiempo. Registro de interrupciones. 	<ul style="list-style-type: none"> Herramienta de registro de tiempos de desarrollo, interrupciones, y <i>Bugs</i> (Figura 52).
WorkBook PSP (Provista por el SEI)	S	S	<p>Se realizan ajustes a la configuración interna de los procesos, eliminando las fases no incluidas como revisiones de diseño, por otra parte se modifica la tabla de medidas para ajustarla a las necesidades de la organización (Tabla 44)</p>	<ul style="list-style-type: none"> Se agrega esta herramienta al proceso ya que posee bastantes funcionalidades y cubrimiento a todos los niveles de PSP ya usados actualmente en el mercado, además por cuestiones de tiempo la modificación de la herramienta propia no fue posible.
UCC tool version 2011.10. (University of Southern California)	S	S	<p>Internamente la herramienta no se modifica pero si se añade una interfaz de usuario que permita un fácil uso al desarrollador sin tener que acceder a la línea de comandos., (Figura 53)</p>	<p>Se agrega esta herramienta con el fin de proveer un elemento automático que permitan realizar el conteo de líneas de código de manera rápida para los desarrolladores</p>
Indigo (Infragistics)	S	N		<ul style="list-style-type: none"> Desarrollo de prototipos funcionales.
Enterprise Architect (Sparx Systems)	S	N		<ul style="list-style-type: none"> Herramienta utilizada para la elaboración de documentación específica para tareas (Casos de uso, Diagramas de Bases de datos, Casos de prueba, etc)
McCabe IQ (McCabe) Trial	S	N		<ul style="list-style-type: none"> Herramienta utilizada para realizar pruebas de calidad sobre el software especialmente en el área de la mantenibilidad.

Tabla 44. Herramientas ajustadas y agregadas al proceso integrado de SCRUM-P
Fuente: Elaboración propia

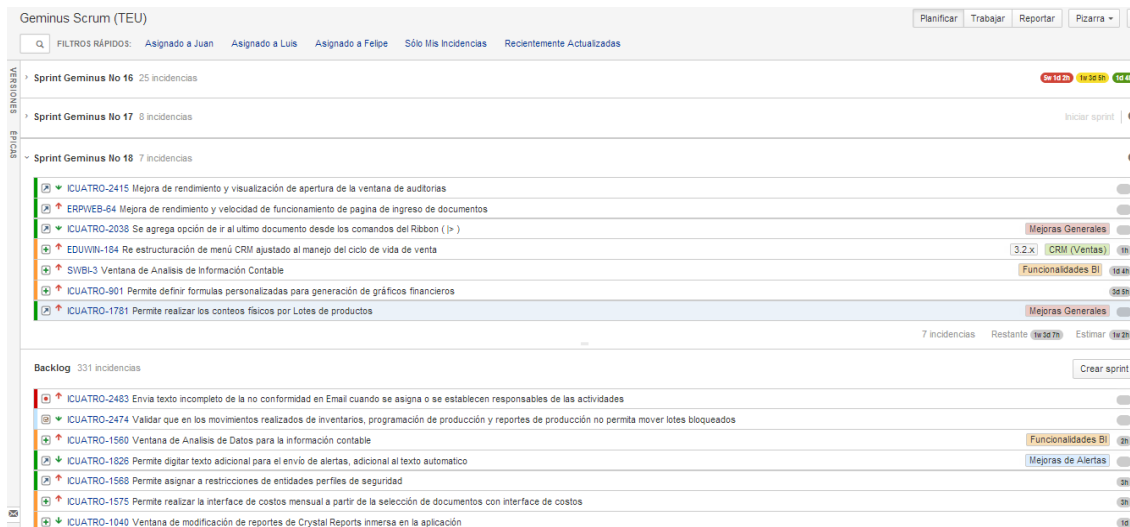


Figura 52. Pantalla Herramienta JIRA para definir Product Backlog

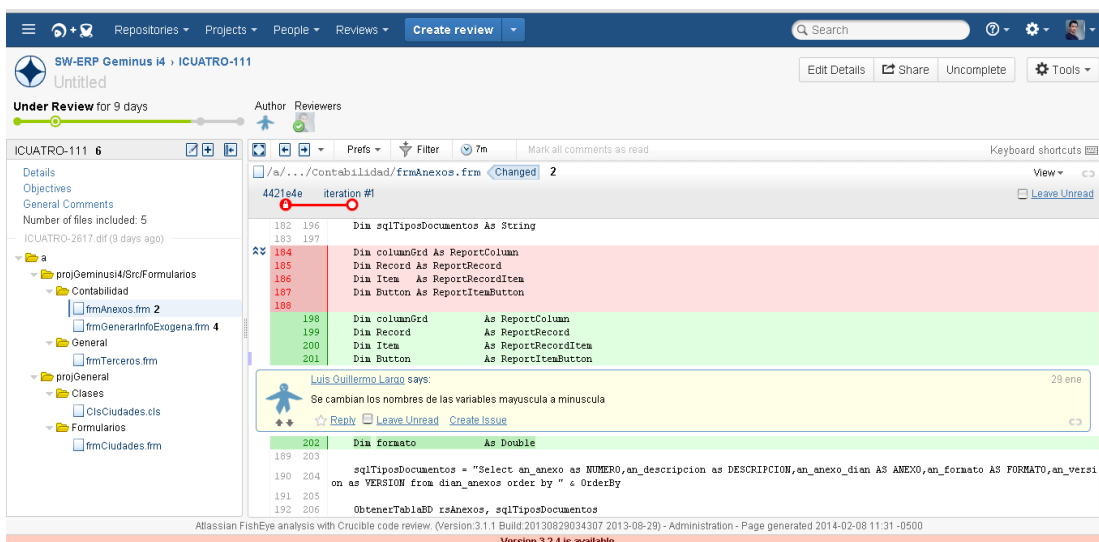


Figura 53. Pantalla de ejemplo herramienta Fisheye

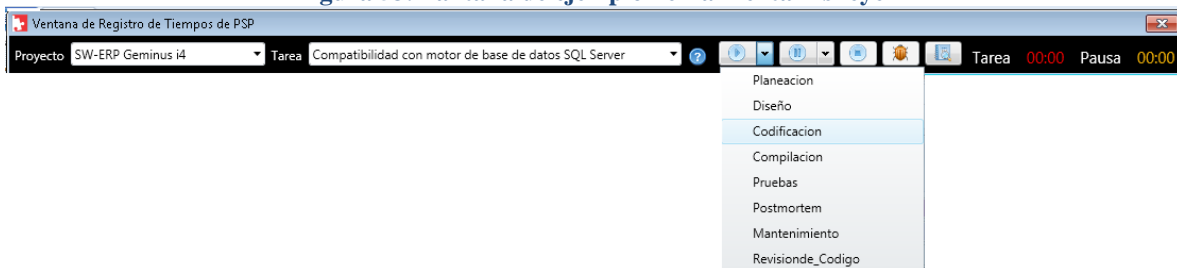


Figura 54. Herramienta de PSP propietaria, Fuente: Elaboración propia

La siguiente tabla corresponde a la modificación de la tabla por defecto de tamaños relativos de la métrica provista en el *Workbook* de PSP del SEI, ya que se ajustó de acuerdo a las necesidades de la organización (Humphrey, Watts S., 2005).

CATEGORÍA	LOC				
	VS	S	M	L	VL
CRUD/Básico	259	511	718	3738	10885
CRUD/Cabera Detalle	294	794	1306	1596	2248
CRUD+Procesamiento	1227	3609	4742	6776	22458
Procesamiento	160	442	737	2817	5062
Reportes	359	736	1059	1954	2872
Búsquedas	237	306	611	972	1740
Clases CRUD	199	277	328	483	2554
Clases Procesamiento	71	179	416	560	1268
Configuración	654	804	1432	2049	6193

Tabla 45. Métricas de medida PSP modificada, Fuente: Elaboración propia

A continuación se relaciona como se elaboró la tabla:

- Obtención de lista de elementos a clasificar: Se obtiene un listado de elementos (Formularios, Clases) elegidos por el equipo que potencialmente pueden servir de referencia para la construcción de la tabla.
- Categorización de proxys: Se clasifican los elementos de acuerdo a las siguientes categorías:

Categoría	Descripción
CRUD/Básico	Formulario que realiza operaciones CRUD.
CRUD/Cabera Detalle	Formulario que realiza operaciones CRUD con una información de cabecera y otra de detalle.
CRUD+Procesamiento	Formulario que realiza operaciones CRUD y que además realiza procesos adicionales con los datos.
Procesamiento	Formulario que realiza operaciones de procesamiento de datos únicamente.
Reportes	Formulario que genera informes de usuario.
Búsquedas	Formularios de tipo búsqueda para el usuario.
Clases CRUD	Clases asociadas a los formularios CRUD, realizan las operaciones en la base de datos.
Clases Procesamiento	Clase que realiza operaciones de procesamiento.
Configuración	Formularios de configuración de la aplicación

**Tabla 46. Clasificación de proxys para elaboración de tabla relativa de tamaños
Fuente: Elaboración propia**

- Obtención de tamaños y división en tallas: Una vez se tienen clasificados los elementos se procede a realizar un conteo de líneas de código y se organizan por los tamaños dados por PSP (VS, S, M, L, VL), ordenándolo de acuerdo a su tamaño a continuación se muestra una tabla de ejemplo de este proceso:

CRUD/Básico	LOC	KLOC	TAMAÑO
frmAdministracionLotes	259	0.259	VS
frmClasificacionesCertificado	409	0.409	
frmSistema	511	0.511	S
frmEstados	585	0.585	
frmCentrosCosto	718	0.718	M
frmEmpresas	1533	1.533	
frmAsistenteCreacionTiposDocs	3738	3.738	L
FrmProductos	6828	6.828	
FrmTerceros	10885	10.885	VL

Tabla 47. Ejemplo de organización de proxys y clasificación por tamaño
Fuente: Elaboración propia

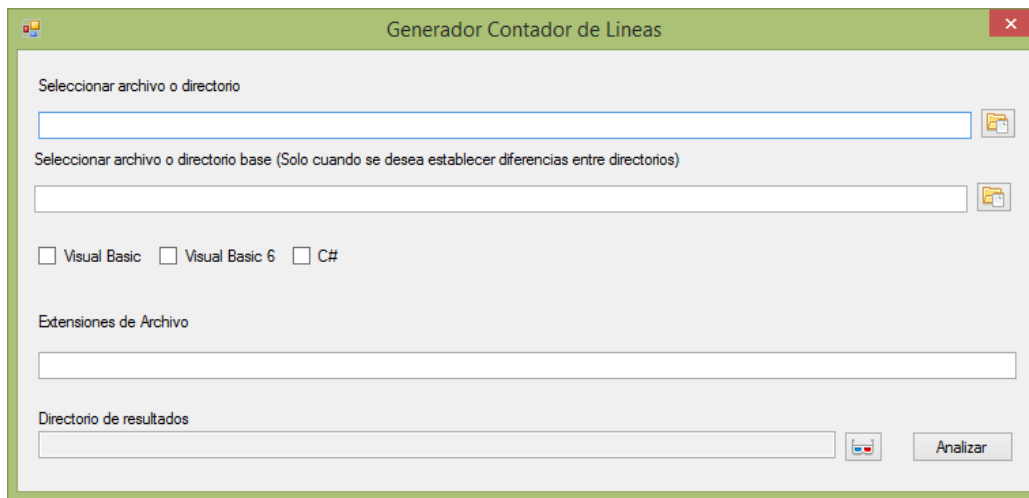


Figura 55 Interfaz de usuario agregada a la herramienta UCC LOC, Fuente : Elaboración propia

6.3.2. RESULTADOS DEL *SPRINT*

En el *sprint* desarrollado en la empresa como prueba piloto de la implementación del modelo integrado entre SCRUM-P y el modelo de calidad ISO, se encontraron los siguientes resultados arrojados de las medidas tomadas durante el desarrollo del mismo.

No de Tareas			
Planeadas	Desarrolladas	Faltantes	% Completado
23	20	3	87%

Tabla 48. Numero de tareas planeadas vs completadas

Se puede observar que se logró completar un porcentaje mayor de tareas, en comparación a los anteriores *Sprint*, en los cuales el promedio de porcentaje completado era del 65.78%

contra un 88% en el actual, aunque no es una cifra contundente y se deben esperar los resultados de futuros *Sprint*, se ve una mejora en el promedio de tareas completadas, otro síntoma de mejora en el *Sprint* piloto, se puede evidenciar en la gráfica *Burndown* del mismo (Figura 54), donde se puede ver como la línea real de trabajo restante en el sprint (en rojo) , es cercana a la línea guía ideal de trabajo restante del *sprint* (en gris), la cual mejoro considerablemente comparándola con *sprints* anteriores.

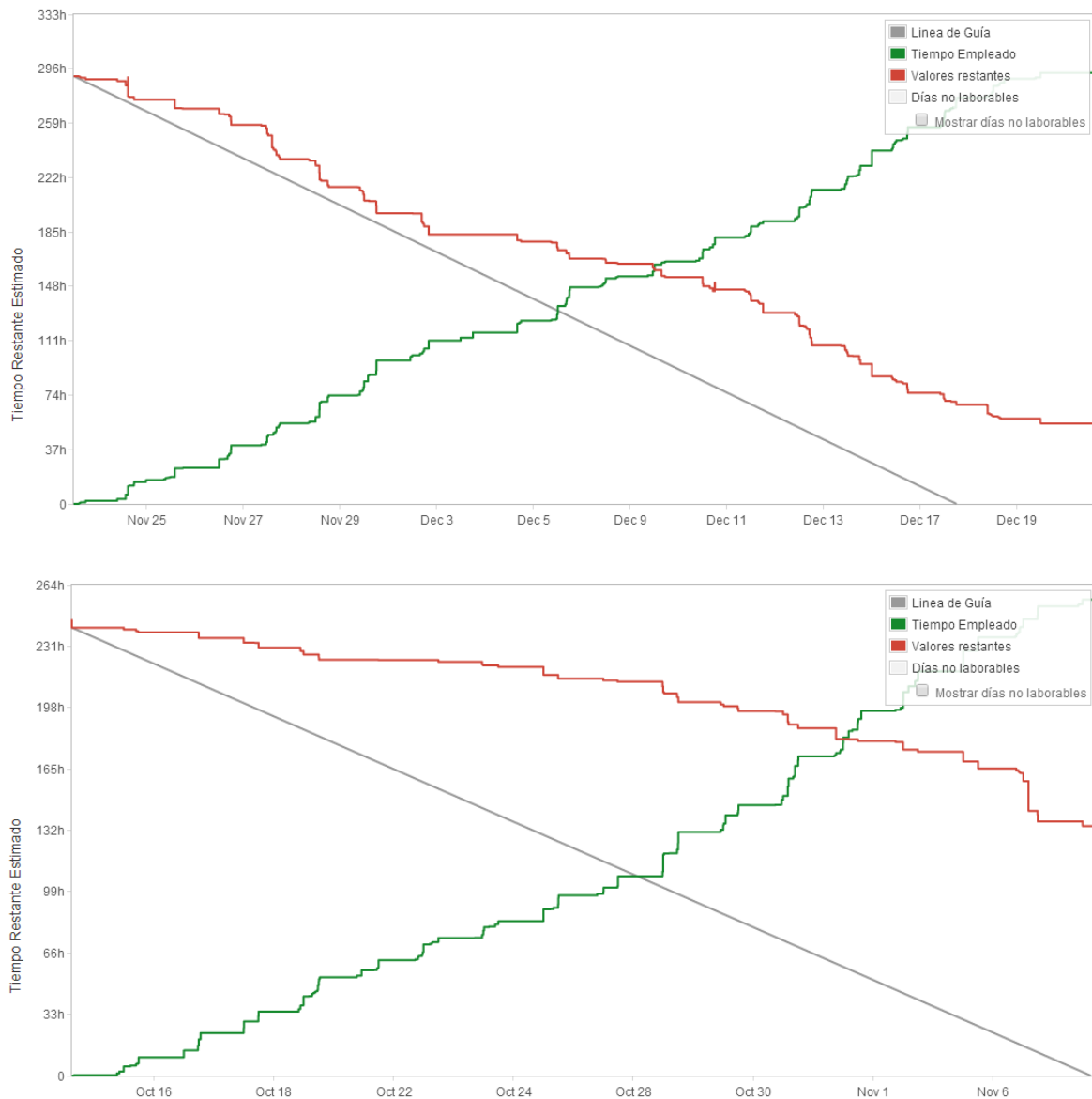


Figura 56. Grafica de *Burndown* chart del Sprint para la prueba piloto

RESULTADOS PSP

Durante el desarrollo del Sprint, cada desarrollador realizó el registro correspondiente a las actividades planteadas para el PSP integrado con SCRUM, para lo cual se hizo uso del PSP *workbook*, herramienta provista por el SEI para este propósito, a continuación se presentan las gráficas provistas por la misma herramienta donde se pueden analizar y ver resultados parciales de la implementación.

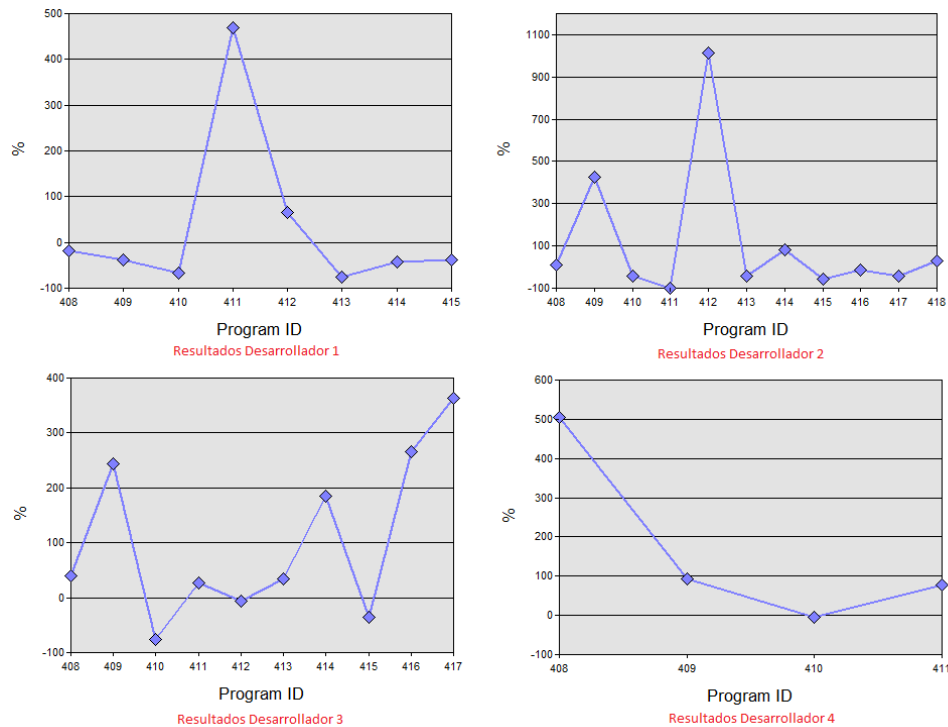


Figura 57. Grafica de error de estimación de tiempo de equipo de desarrollo

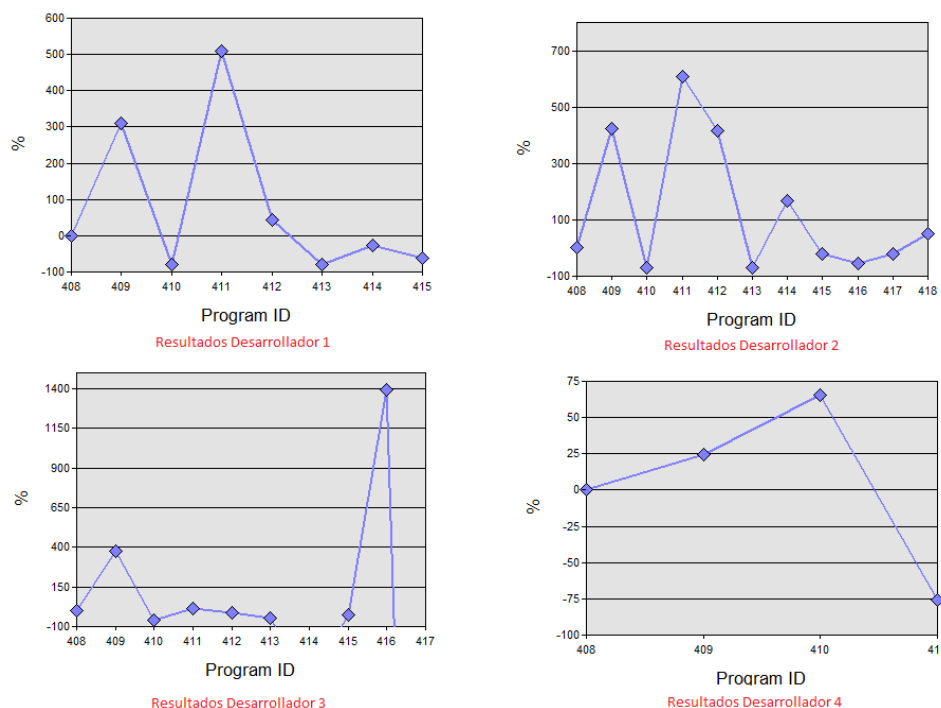


Figura 58. Grafica de error de estimación en tamaño equipo de desarrollo

Durante el desarrollo del *sprint* cada desarrollador aplico su proceso de PSP registrando los datos en paralelo en la herramienta PSP seleccionada, se puede observar que la gráfica de porcentaje de error de estimación cada vez es más bajo tanto para tamaño como para tiempo para la mayoría de desarrolladores, lo cual favorece al proceso de estimación de tal manera que los *Sprint* sean más exactos y acercados a la realidad, basados en datos históricos y no en la intuición del desarrollador.

ENCUESTA DE PERCEPCIÓN AL EQUIPO DE DESARROLLO

Una vez implementado el proceso integrado de SCRUM-P, se procede a realizar nuevamente una encuesta en el equipo de desarrollo, con el fin de establecer la percepción de la implementación de la prueba piloto con el nuevo proceso integrado, a continuación se presentan los resultados de la encuesta:

Ficha Técnica Encuesta	
Nombre del Estudio	Encuesta percepción de proceso integrado SCRUM-P
Liderada por :	Jhon Alexander Holguín Barrera
Ejecutada y analizada por:	Jhon Alexander Holguín Barrera
Instrumento:	Cuestionario de 8 Preguntas
Marco de la muestra:	Equipo de Desarrollo de la organización
Metodología:	Cuantitativa

Muestreo:	Todo el Equipo
Técnica:	Encuesta virtual a través de la aplicación Survey Monkey
Fecha de realización:	28 de Enero de 2014
Tamaño de la muestra:	5 Personas

Tabla 49 Ficha técnica encuesta desarrolladores posterior a la implementación de SCRUM-P

PREGUNTAS Y RESULTADOS DE LA ENCUESTA

1. ¿Considera que existe una mejor integración de SCRUM y PSP con respecto al proceso manejado anterior?

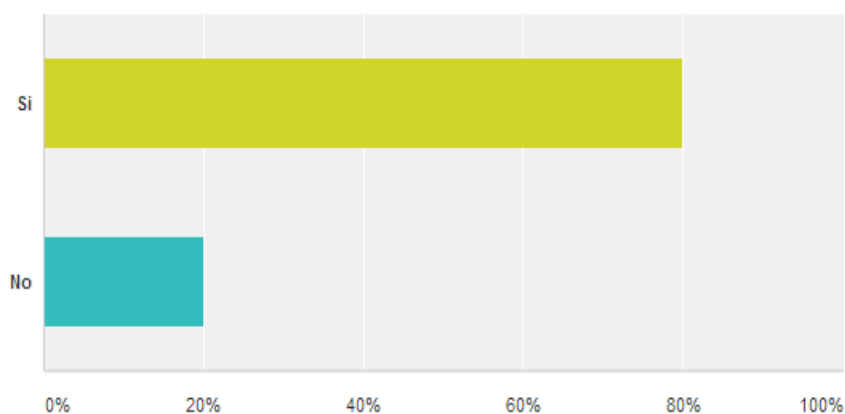


Figura 59. Resultados, Considera que existe una mejor integración de SCRUM y PSP con respecto al proceso manejado anterior?

2. Considera que las actividades definidas en el proceso de SCRUM y PSP integrado:

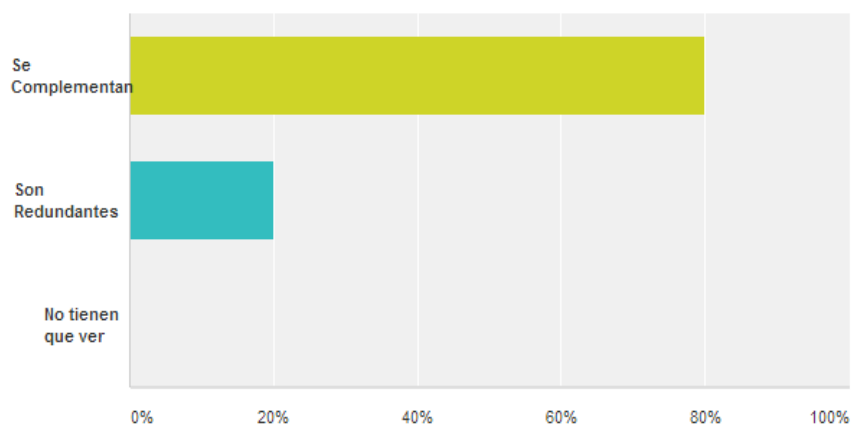


Figura 60. Resultados, Considera que las actividades definidas en el proceso de SCRUM y PSP integrado

3. Considera que el proceso de SCRUM y PSP integrado está

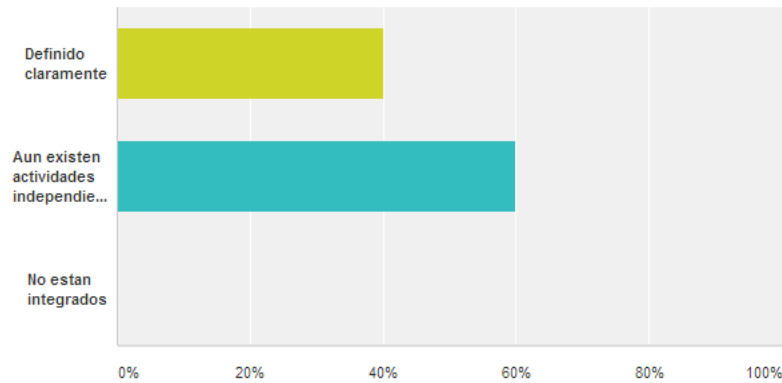


Figura 61. Resultados, Considera que el proceso de SCRUM y PSP integrado esta

4. Considera que la estimación guiada por PSP con respecto a la actividad del Sprint *Planning* en SCRUM

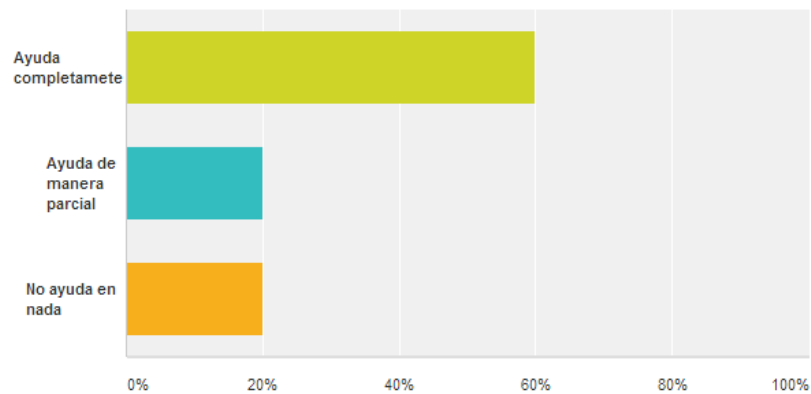


Figura 62. Resultados, Considera que la estimación guiada por PSP con respecto a la actividad del Sprint Planning en SCRUM

5. Considera que el proceso integrado de SCRUM y PSP impacta el proceso de desarrollo

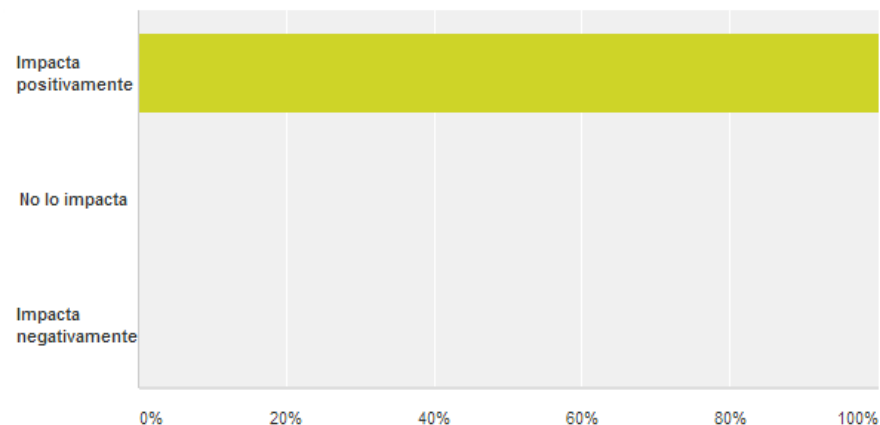


Figura 63. Resultados, Considera que el proceso integrado de SCRUM y PSP impacta el proceso de desarrollo

6. Considera que PSP integrado con SCRUM impacta el control de calidad del desarrollo

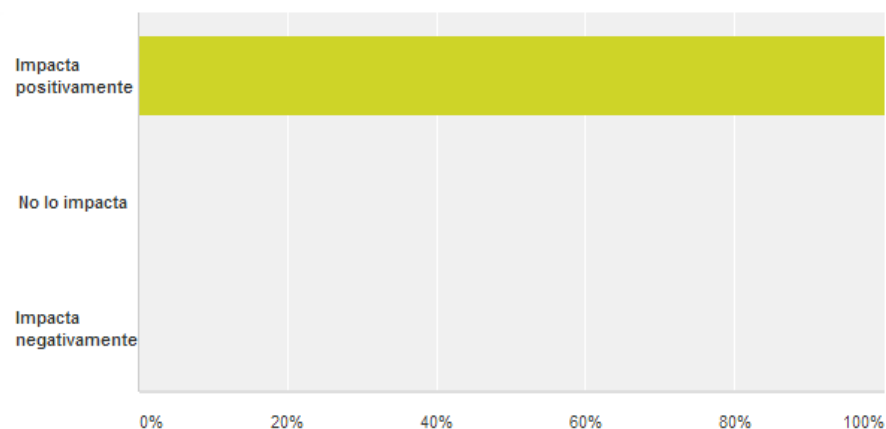


Figura 64. Resultados, Considera que PSP integrado con SCRUM impacta el control de calidad del desarrollo

7. ¿Considera la aplicación del proceso integrado de SCRUM y PSP dentro del proceso de desarrollo, favorece al ambiente de trabajo del equipo?

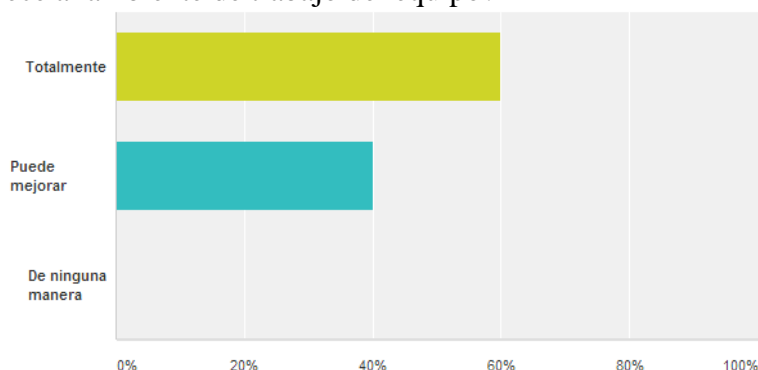


Figura 65. Resultados, Considera la aplicación del proceso integrado de SCRUM y PSP dentro del proceso de desarrollo, favorece al ambiente de trabajo del equipo?

8. ¿Qué recomendaciones puede dar para mejorar el proceso de desarrollo que actualmente se maneja en la organización?

Respuestas
Que exista un poco más de documentación para cada tarea y del "negocio" en general
Permitir al equipo de desarrollo ser parte del diseño de las tareas
Como las personas son reacias al cambio, se necesita acabar de tomar conciencia que estos procesos son para mejorar
Que las presiones é los clientes o las tareas "Urgentes", no desplacen los procesos que se deben realizar
Una capacitación más abierta sobre el tema
Evaluar a través de las retrospectivas los resultados de los sprint y el proceso integrado

Como se puede observar la percepción del equipo sobre el proceso ha mejorado, aunque aún existen puntos por mejorar y que deben ser tenidos en cuenta para tener un mejoramiento continuo en el proceso. Dentro de los resultados de las preguntas del impacto

de la integración en el proceso y la calidad, se puede observar que existe una unanimidad sobre la percepción de que esta impacta de una manera positiva en ambos puntos.

6.3.3. RESULTADOS DE CALIDAD DEL INCREMENTO DEL SPRINT

De acuerdo a las características de calidad seleccionadas en el modelo ISO, se realizó el proceso de medición con el fin de establecer la calidad del producto en el incremento del *Sprint* (no se evaluara la calidad total del producto si no solo de las funcionalidades que se liberaron en el *Sprint*), para esto se establecieron los valores mínimos de cumplimiento con el equipo y el propietario del producto, a continuación se presentan los resultados de acuerdo a cada una de las categorías.

FUNCIONALIDAD

Para la evaluación de la categoría de funcionalidad se realizó un listado de las tareas programadas durante el *Sprint* y se evaluaron de acuerdo a las métricas definidas (Anexo 6) anteriormente en el modelo de calidad; a continuación se presentan los resultados:

FUNCIONALIDAD					
Meta	Preguntas	Métricas	Valor Resultado	Valor Esperado	Cumple (S/N)
Evaluar el cumplimiento de los requerimientos definidos en un <i>Sprint</i>	¿Se implementaron todos los requerimientos en el <i>Sprint</i> ?	Porcentaje de requerimientos que se entregaron en el <i>Sprint</i> con respecto al total planeados	86.95%	>=80%	S
	¿Los requerimientos desarrollados cumplen su función?	Porcentaje de desarrollos que cumplen con todas las expectativas descritas en el requerimiento	90%	>=90%	S
	¿Los requerimientos son exactos en la información presentada al usuario?	Porcentaje de requerimientos entregados en el <i>Sprint</i> que tuvieron errores en datos o funcionamiento durante la fase de pruebas con respecto al total de requerimientos desarrollados	15%	<=5%	N

Tabla 50. Resultados mediciones criterios de calidad de funcionalidad

Fuente: Elaboración propia

RENDIMIENTO

Para el desarrollo de las pruebas de rendimiento, se toma el producto resultante del incremento del *Sprint* y se seleccionan las tareas más críticas (de acuerdo al criterio del *product owner*) para realizar las pruebas. Se utiliza el administrador de tareas de Windows y se realizan las mediciones (Anexo 7) con los resultados mostrados a continuación:

RENDIMIENTO					
Meta	Preguntas	QMEs	Valor Resultado	Valor Esperado	Cumple (S/N)
Establecer la eficiencia	¿Qué tan eficiente son	Porcentaje de tareas del total que	83.33%	>=80%	S

de las funcionalidades implementadas en el <i>Sprint</i>	las funcionalidades desarrolladas en el <i>Sprint</i> ?	no sobrepasan el tiempo definido para completar el trabajo			
	¿Cuánto procesamiento utiliza una funcionalidad para ser finalizada?	Porcentaje de tareas del total que no sobrepasan el Porcentaje de procesamiento definido desde el inicio a la finalización de una tarea	33.33%	>=50%	N
	¿Cuánto espacio de memoria es utilizado para finalizar una tarea dada?	Porcentaje de tareas del total que no sobrepasan el porcentaje de consumo de memoria definido desde el inicio de la tarea hasta la finalización	83.33%	>=80	S

Tabla 51. Resultados mediciones criterios de calidad de rendimiento
Fuente: Elaboración propia

USABILIDAD

Para el desarrollo de las pruebas de usabilidad también se seleccionaron las funcionalidades más críticas a juicio del propietario del producto y se realizan las pruebas con un usuario normal que posee conocimientos básicos de la aplicación, se le indica que ejecute las funcionalidades y se toman las mediciones correspondientes (Anexo 8), a continuación se relacionan los resultados:

USABILIDAD					
Meta	Preguntas	QMEs	Valor Resultado	Valor Esperado	Cumple (S/N)
Establecer la usabilidad de las funcionalidades implementadas en el <i>Sprint</i>	¿Cuánto tiempo tardan los usuarios en aprender a usar una funcionalidad?	Porcentaje de tiempo adicional requerido del estimado para el aprendizaje de uso de la funcionalidad	33.75%	Promedio <=20%	N
	¿Qué proporción de funciones requieren demostración?	Porcentaje de funcionalidades implementadas sobre el total que requieren de una demostración	25%	<=30%	S
	¿Qué proporción de las funcionalidades son correctamente descritas en la documentación de usuario o ayuda?	Porcentaje de funcionalidades que poseen una documentación o ayuda de usuario	5%	>=80%	N
	¿Cuántos mensajes pueden ser entendidos de manera fácil?	Porcentaje de mensajes entendidos de manera fácil sobre el total de mensajes mostrados al usuario en las implementaciones	95%	>=90%	S

Tabla 52. Resultados mediciones criterios de calidad de usabilidad
Fuente : Elaboración propia

MANTENIBILIDAD

Para el desarrollo de las pruebas de mantenibilidad se utilizó la herramienta McCabe IQ (Versión Trial) a través de la cual se realizaron pruebas automatizadas sobre el código que

permitan obtener las mediciones requeridas para las pruebas de calidad; se realizó una selección aleatoria del código fuente agregado (se utiliza solo el código fuente nuevo para determinar qué tan mantenible es, a este se le realizaron las respectivas pruebas a través de la herramienta (Anexo 9), a continuación se muestran los resultados:

MANTENIBILIDAD					
Meta	Preguntas	QMEs	Valor Resultado	Valor Esperado	CUMPLE (S/N)
Que tan mantenible es el producto	¿Qué capacidad tiene el código de ser analizado?	Porcentaje de las funciones agregadas del total que superan la complejidad ciclomática establecida	4.63 %	<=5%	S
		Porcentaje de líneas de código que poseen comentarios sobre el total de líneas de código lógicas	15.31 %	>=40%	N
	¿Existen probabilidades de errores en el código por complejidad ciclomatica?	Porcentaje de elementos del total que poseen probabilidad del 20% de tener problemas (Según la tabla McCabe)	33.33%	<=10%	N
	¿Qué tan reutilizable es el código?	Porcentaje del total de funcionalidades que cumplen con que deben tener mínimo un elemento reutilizable (Clase)	50%	>=70%	N

Tabla 53. Resultados mediciones criterios de calidad de mantenibilidad

Fuente : Elaboración propia

De acuerdo a la información presentada por la herramienta se obtuvieron los resultados anteriores, aun así se quiere resaltar que la herramienta posee múltiples informes gráficos que se presentan a continuación y que permiten dar más aun una visibilidad de la calidad del producto.

A continuación se presenta una gráfica que permite visualizar la complejidad para realizar pruebas sobre una función desarrollada por un programador en la cual se expone su dificultad por los posibles caminos existentes para la realización de las pruebas, esta funcionalidad fue una de las que obtuvo mayor puntaje de complejidad ciclomática (47 Muy compleja y de alto riesgo).

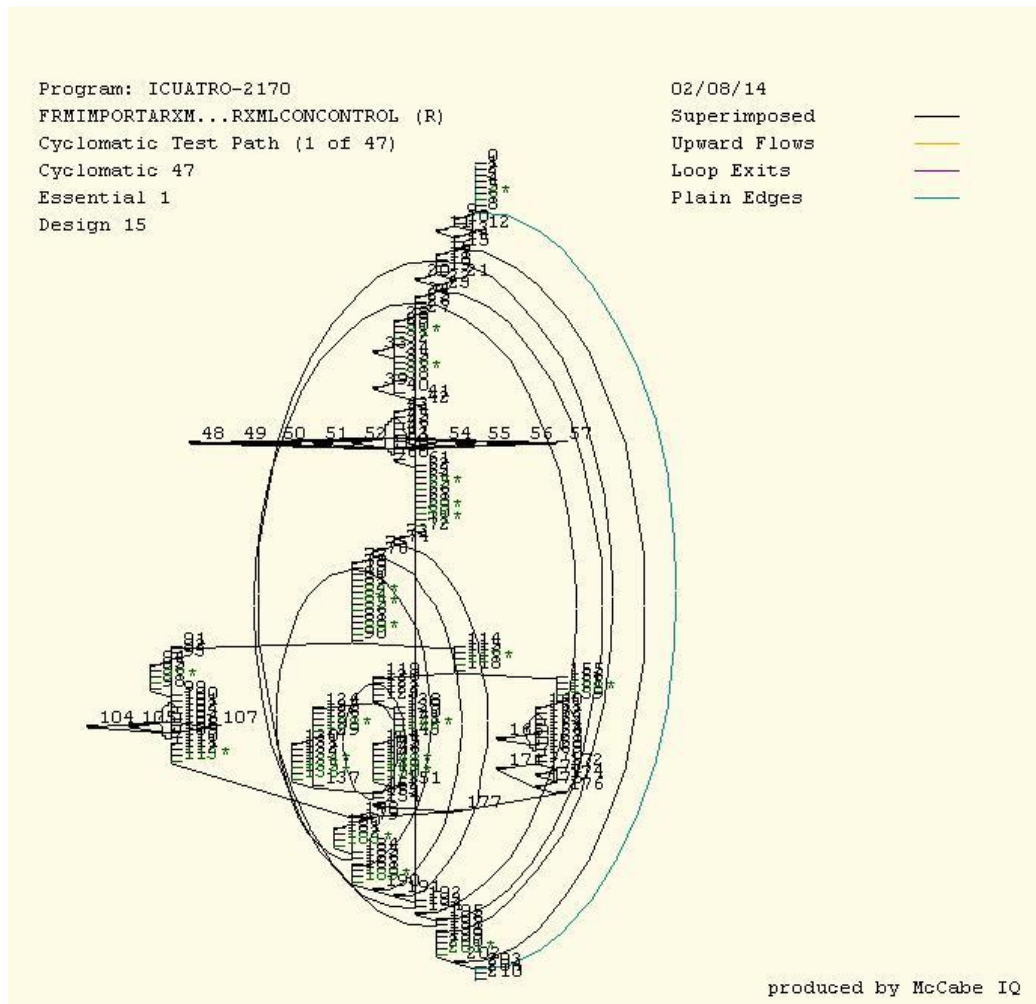


Figura 66. Ruta de pruebas generado por la herramienta McCabe (McCabe Software, 2014) para la funcionalidad con mayor complejidad ciclomática (47)

A continuación se presenta una gráfica tipo Kiviat, que permitió analizar el código fuente de una funcionalidad desarrollada en el *Sprint* donde se puede ver desde las perspectivas de mantenibilidad, capacidad de ser probado y tamaño.

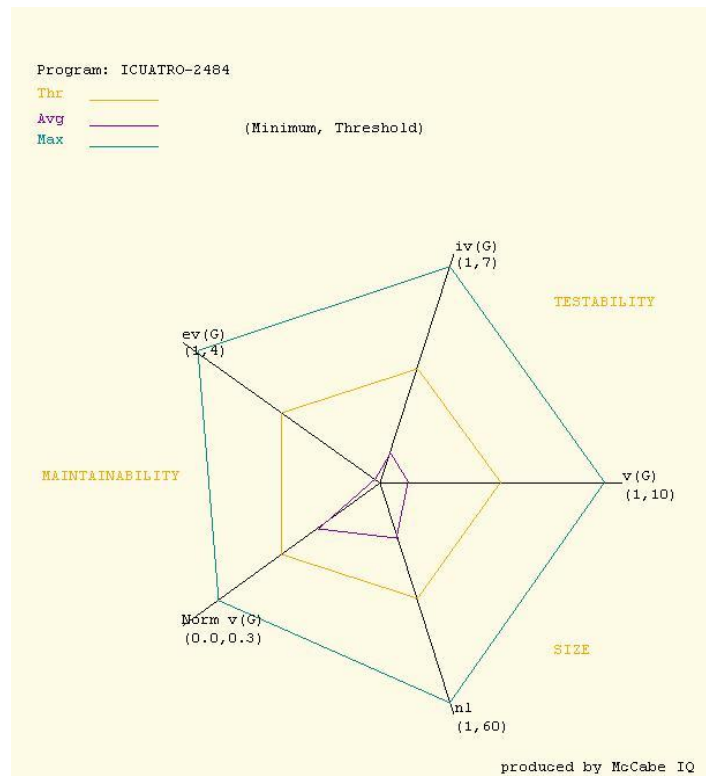


Figura 67. Grafico Kiviatic de funcionalidad del Sprint producido por McCabe IQ

RESUMEN DE RESULTADOS DE CALIDAD

De acuerdo a los resultados obtenidos se procede a realizar una tabla resumen donde se muestra el cumplimiento de cada una de las características seleccionadas de la norma, de acuerdo a cantidad de criterios cumplidos en la evaluación, permitiendo tener de manera visual un resultado global del cumplimiento de los estándares de calidad del producto establecidos.

CARACTERÍSTICA	PUNTOS EVALUADOS	PUNTOS OK	PUNTOS NO OK	% OK
FUNCIONALIDAD	3	2	1	67%
USABILIDAD	4	2	2	50%
RENDIMIENTO	3	2	1	67%
MANTENIBILIDAD	4	1	3	25%
TOTAL	14	7	7	52%

Tabla 54. Resumen de resultados mediciones criterios de calidad, Fuente: Elaboración propia

Gracias a la tabla anterior se puede obtener una perspectiva sobre la calidad actualmente manejada en el producto, la cual está basada en la evaluación de las características de la norma seleccionadas, en donde se puede apreciar que es necesario mejorar en aspectos como la mantenibilidad y la usabilidad, y esto a su vez se reflejará en la calidad global del producto. Otro aspecto importante es poder reflejar que a pesar de que se tenga un proceso organizado, es necesario realizar una evaluación más objetiva sobre el producto que permita establecer el estado de calidad del producto, permitiendo establecer cuáles son los puntos fuertes y débiles a los cuales debe colocarse atención.

Debido a que no existen resultados previos de evaluación de calidad de producto, no fue posible establecer un punto de comparación con los resultados arrojados.

IT MARK

Durante el proceso de integración de SCRUM-P y su implementación en la prueba piloto, la organización inició un proceso de evaluación de IT MARK haciendo uso de recursos dados por el estado para este fin esto en colaboración con la Universidad Tecnológica de Pereira. Esta evaluación estaba dividida en dos fases en las cuales la primera era una revisión para establecer puntos de mejora y la segunda fase correspondía a la reevaluación con las correcciones realizadas de acuerdo a las apreciaciones de la evaluación.

Durante la primera evaluación, se presentaron los modelos integrados de SCRUM-P, para la evaluación correspondiente a desarrollo en IT MARK (CMMI-Nivel 2 SCAMPI clase C (CMMI INSTITUTE, 2014)), obteniendo los siguientes resultados en cada área de proceso.

<u>ÁREA</u>	<u>DESCRIPCIÓN</u>	<u>PUNTAJE OBTENIDO</u>	<u>OBSERVACIONES</u>
REQM	GESTIÓN DE REQUISITOS	90%	
PP	PLANIFICACIÓN DE PROYECTO	85.04%	
PMC	SEGUIMIENTO Y CONTROL DE PROYECTO	82.04%	
PPQA	ASEGURAMIENTO DE LA CALIDAD DE PROCESO Y PRODUCTO	30.44%	No se obtuvo un buen puntaje debido a que no se está haciendo seguimiento en el cumplimiento del proceso "Auditorias"
MA	MEDICIÓN Y ANÁLISIS	87.3%	
CM	GESTIÓN DE CONFIGURACIÓN	22.5%	No se obtuvo un buen puntaje ya que solo se está gestionando el código en gestión de configuración y no todo el proyecto.
Nivel de madurez 2		66.22%	

Tabla 55. Resultados de evaluación IT Mark, CMMI Nivel 2
Fuente: Informe de Resultados evaluación IT Mark

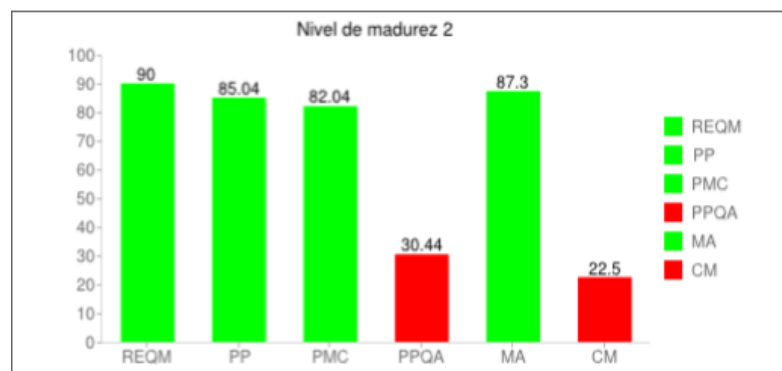


Figura 68. Grafica de Resultados evaluación IT-Mark

Fuente: Informe de Resultados evaluación IT Mark

Como se puede observar el proceso integrado de SCRUM-P favoreció de manera positiva el proceso de evaluación de IT MARK en la organización, permitiéndole obtener el 66.22 % de nivel de madurez de CMMI nivel 2 y así permitirle obtener la certificación en la primera evaluación del proceso, siendo la única organización que logró esta certificación en la primera fase de las empresas participantes en el proceso; esto permite resaltar que el proceso favorece al cumplimiento de los estándares definidos para la calidad de software para los entes que regulan estas actividades.

DIFICULTADES EN EL PROCESO

Durante todo el proceso se pudieron encontrar diferentes tipos de dificultades que deben ser tenidas en cuenta para próximas implementaciones y que probablemente permitan obtener mejores resultados del mismo, algunas de estas que se rescatan son:

- **Entrenamiento PSP:** Una de las desventajas que se encontró es que el hecho de que los miembros del equipo no tuvieran una capacitación o curso de entrenamiento formal en PSP, hizo que el proceso fuera lento y con algunos errores en el ingreso de la información, causando re-procesos y demoras en el análisis de los datos.
- **El análisis de calidad aplicando los criterios seleccionados** es un proceso que requiere de recursos (Personal, Tiempo, Herramientas), para lo cual realizarlo después de cada *Sprint* puede ser costoso, es necesario evaluar si es necesario modificar el intervalo de tiempo de evaluación o el número de implementaciones o desarrollos a ser tenidos en cuenta en la evaluación de calidad.
- **La resistencia de algunos miembros del equipo a la implementación de la integración de SCRUM y PSP**, es un factor que debe anticiparse para evitar que se presente y contagie a los demás miembros del equipo, es necesaria una alta disposición y concientización del equipo, para que ponga todo de su parte y lograr aplicar el proceso de manera

disciplinada, lo cual puede mejorar una vez cada desarrollador se da cuenta de su rendimiento y productividad.

- Existieron adecuaciones al proceso que fueron necesarias hacerlas durante la implementación, causando duda y recapitaciones con el fin de lograr comunicar los cambios.
- La disciplina de la captura de *Bugs* fue una de las dificultades más críticas en el proceso de PSP que debió corregirse en varias ocasiones, esto debido a que es una disciplina que se adquiere con la práctica.
- Las pruebas de calidad llevaron demasiado tiempo, especialmente las de rendimiento, además sacar los indicadores de cumplimiento de las diversas características fue un factor que no era muy claro y tuvo que hacerse a juicio de expertos inicialmente, es necesario establecer una política de calidad clara para saber cuáles son los lineamientos a cumplir y además de esto tener información histórica que permita evaluar el grado de mejora de calidad.

RECOMENDACIONES

A continuación se realizan algunas anotaciones de puntos a tener en cuenta que pueden permitir facilitar la implementación del proceso de una manera más eficaz.

- Herramientas: Una de las ventajas que se pudo encontrar en la implementación del proceso integrado, fue que se contaba con herramientas tales como el JIRA, Fisheye, PSP (Propietario), PSP *Workooook* las cuales facilitaban el desarrollo de las actividades de cada proceso y de esta manera desgastar menos tiempo en el ingreso de datos que el desarrollo de tareas además estas herramientas cuentan con costos muy bajos o gratuitos para empresas que están iniciando.
- Divulgación del proceso: Tener claro desde el principio el proceso que se va a implementar, como se integra con las herramientas con las que se cuenta y divulgarlo al equipo juega un factor decisivo a la hora de la implementación del mismo, ya que permite dar seguridad al equipo y al líder de cuáles son los pasos o acciones en cada uno de los procesos, evitando recapitaciones o preguntas frecuentes sobre el mismo.
- Retrospectivas: La implementación del PIP en PSP y las retrospectivas en SCRUM, son elementos clave dentro del proceso que permiten tener un mejoramiento continuo y auto dirigido, ya que cada miembro y el equipo tiene la capacidad de autoevaluar su proceso y modificar en caso de ser necesario.

- La evaluación de planeación de PSP en el *Sprint planning* requiere que exista una documentación aunque no exhaustiva si detallada de lo que se desea hacer, de tal manera que permita al desarrollador estimar de acuerdo a sus datos históricos el esfuerzo requerido para finalizar la tarea.

6.4. CONCLUSIONES

- Se puede evidenciar que SCRUM y PSP son elementos dentro del proceso de desarrollo que se complementan entre sí, a través de las ventajas y características que cada uno de los modelos ofrece, SCRUM como marco de trabajo que permite realizar una gestión eficiente del proceso de desarrollo de software, desde la entrada del requerimiento hasta su salida (entrega al cliente) a nivel de equipo, y PSP como modelo para potencializar la capacidad de cada desarrollador para gestionar su trabajo con calidad, permitiendo obtener métricas y control del proceso a nivel individual.
- La utilización de las métricas para estimación de tamaño y tiempo en PSP, son una herramienta efectiva y que ayuda al proceso de planeación de SCRUM, dando al equipo una mayor precisión ya que se basa en datos históricos reales provenientes de la información ingresada por cada miembro del equipo.
- La integración de SCRUM y PSP, es una herramienta que busca impactar positivamente a la calidad del producto, pero aun así no es garantía de que el producto sea de calidad; es por esto que la integración del modelo de calidad ISO para el producto es una herramienta potente que permite complementar el proceso para establecer y medir los parámetros de calidad que la organización establezca.
- Uno de los factores de éxito de la integración de SCRUM y PSP fue el uso de herramientas que ayuden a minimizar el impacto de la implementación de las actividades definidas para el proceso, de tal manera que no centren el esfuerzo de los miembros del equipo en la recolección de la información y gestión del proceso.
- GQM es un modelo que permite definir métricas de manera fácil y clara que permitan establecer la meta y como llegar a ella, en el presente proyecto se utilizó para la definición de las métricas de calidad de la norma ISO, pero puede extenderse al resto del proceso en una mejora futura del mismo.
- Otro elemento relevante a tener en cuenta dentro de los resultados encontrados, fue como la definición del proceso integrado de SCRUM y PSP ayudó de manera determinante a la organización en la obtención de la certificación IT Mark, sin estar está planeada dentro del proyecto, esto permite evidenciar que este modelo permite de

igual forma cumplir con los estándares internacionales para la mejora de procesos como lo es el propuesto por el ESI a través de IT Mark.

7. BIBLIOGRAFÍA

- Abran, A., Al Qutaish, R., Desharnais, J.-M., & Habra, N. (2005). An Information Model for Software Quality Measurement with. *Proceedings of the International Conference on Software Development (SWDC'05)*, (págs. 104-116). Reykjavik. Obtenido de ResearchGate.
- Abud Figueroa, M. A. (s.f.). *Calidad en la Industria del Software. La Norma ISO-9126*.
- Albaladejo, X. (18 de Febrero de 2012). Introducción a Agile y Lean. España.
- Beedle, M., Van Bennekum, A., Cockburn, A., Ward, C., Fowler, M., Highsmith, J., . . . Thomas, D. (13 de 11 de 2001). *Manifiesto por el Desarrollo Ágil de Software*. Recuperado el 05 de 09 de 2013, de <http://www.agilemanifesto.org/iso/es/principles.html>
- Booch, G. (1995). *Object Solutions: Managing the Object-Oriented Project*. Addison-Wesley.
- Calero, C., Moraga, M. A., & Piattini, M. G. (2010). *Calidad del producto y proceso software*. Madrid: Editorial Ra-Ma.
- Chrissis, M. B., Konrad, M. D., & Shrum, S. (2010). *CMMI for Development: Guidelines for Process Integration and Product Improvement ; [CMMI-DEV, Version 1.3]*. Addison Wesley Professional.
- CMMI INSTITUTE. (10 de 09 de 2014). *CMMI Appraisal Classes*. Obtenido de <http://cmmiinstitute.com/cmmit-solutions/cmmit-appraisals/cmmit-appraisal-classes/>
- Cortés Morales, R. (1998). *Introducción al Análisis de Sistemas y la Ingeniería de Software*. Costa Rica: EUNED.
- Crosby, P. B. (1996). *The Quality is Free*. the University of Michiga: McGraw-Hill.
- Cusumano, M., MacCormack, A., Kemerer, C. F., & Crandall, W. (2003). *A Global Survey of Software Development Practices*. MITSloan.
- Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). *The Scrum Primer 2.0*. Obtenido de Scrum Foundation: <http://assets.scrumfoundation.com/downloads/1/scrumprimer20.pdf?1352449266>
- Digital, C. (13 de 03 de 2007). *¿Hacia dónde debe apuntar el software colombiano?* Recuperado el 21 de 04 de 2012, de El Tiempo.COM Blog: http://www.eltiempo.com/participacion/blogs/default/un_articulo.php?id_blog=3349709&id_recurso=400000318
- Fedesoft, MINTIC, Vive Digital FiTi. (Noviembre de 2012). *Estudio de la caracterización de productos y servicios de la industria de software y servicios asociados*. Recuperado el 20 de Junio de 2013, de Fedesoft: <http://fedesoft.org/estudio-de-la-caracterizacion-de-productos-y-servicios-asociados-2012/>
- Griful Ponsati, E., & Canela Campos, M. Á. (2010). *Gestión de la calidad*. Cataluña: Univ. Politèc. de Cataluña.
- Grupo ISSI. (2003). *Metodologías Ágiles en el Desarrollo de Software*. Alicante: Grupo ISSI.
- GTO Partners in Performance. (25 de 04 de 2007). *Informatica y Calidad*. Recuperado el 27 de 04 de 2012, de http://www.itq.ch/pdf/sqm/07/sqm_moths_pr.pdf
- Guoping, R., Dong, S., & He, Z. (2010). SCRUM-PSP: Embracing Process Agility and Discipline. *Software Engineering Conference (APSEC), 2010 17th Asia Pacific*, (págs. 316-325). Sydney, NSW.
- Humphrey, W. S. (1989). *Managing the Software Process*. SEI Series In Software Engineering.
- Humphrey, W. S. (2000). *The Personal Software Process (PSP)*. Carnegie Mellon.

- Humphrey, Watts S. (2005). *PSP(sm): A Self-Improvement Process for Software Engineers*. New Jersey: Pearson Education.
- IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology std 610.12-1990*. IEEE.
- IEEE Computer Society. (2012). *IEEE Standard for Software Quality Assurance Plans*. New York: The Institute of Electrical and Electronics Engineers, Inc.
- IEEE, Computer Society. (2004). *Software Engineering Body of Knowledge -SWEBOK*. Los Alamitos, California: IEEE, Computer Society.
- ISO. (2005). ISO/IEC 25000. En *Software engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE* (pág. 1). ISO/IEC.
- ISO. (2011). *ISO/IEC 25010:2010, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. ISO/IEC.
- ISO/IEC. (2008). *Systems and software engineering — Software life cycle processes ISO/IEC 12207* (Second Edition ed.).
- ISO/IEC. (2009). *Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) – Evaluation reference model and guide*. CELEPAR – Companhia de Informatica do Parana.
- ISO/IEC. (2011). *ISO/IEC 25023*. 31: 08.
- Jeffries, R., Anderson, A., & Hendrickson, C. (2001). *Extreme Programming Installed*. New Jersey: Addison-Wesley.
- Jones, C. (2 de Noviembre de 2010). *SOFTWARE QUALITY IN 2010, A SURVEY OF THE STATE OF THE ART*. Recuperado el 20 de 11 de 2012, de Software Quality Group of New England: <http://www.sqgne.org/presentations/2010-11/Jones-Nov-2010.pdf>
- Kruchten, P. (2003). *The Rational Unified Process an Introduction-Third Edition*. Addison Wesley.
- McCabe Software. (10 de 09 de 2014). *Baseline Code Analysis Using McCabe IQ*. Obtenido de <http://www.mccabe.com/pdf/Baseline%20Code%20Analysis%20Using%20McCabe%20IQ.pdf>
- McHale, J., Chick, T. A., & Miluk, E. (2010). *Implementation Guidance for the Accelerated Improvement Method (AIM)*. Pittsburgh: SEI.
- Pardo, C., Pino, F. J., Garcia, F., & Piattini, M. (2009). Analizando el apoyo de marcos SPI a las características de calidad del producto ISO 25010. *Revista Española de Innovacion, Calidad e Ingenieria del Software*, 5(2), 6-16.
- Perez Queiruga, L., & Vallespir, D. (2009). *Adaptación de RUP para PSP: Modelo de Proceso y Caso de Estudio*. Montevideo: Reporte Técnico RT 09-15.
- Poppendieck, M. (2005). A History of Lean: From Manufacturing to Software Develo. in *JAOO Conference*. Aarhus, Denmark.
- Pressman, R. (2002). *Ingeniería del Software: un enfoque práctico*. Madrid: McGraw-Hill.
- Priolo, S. (2009). *Métodos Ágiles*. Users.
- PROEXPORT. (21 de 11 de 2011). Oportunidades de Negocios en Estados Unidos. *EL TIEMPO*.
- PROEXPORT. (2012). *Abecé del TLC Colombia - Estados Unidos*. Bogota: PROEXPORT COLOMBIA.

- PSL. (01 de 11 de 2013). *PSL - CMMI® 5 Software Development from Latin America - Software Development*. (PSL) Recuperado el 20 de 03 de 2013, de <http://www.pslcorp.com/it-services/application-development.html>
- Rational Company. (1998). *Rational Software Process, Best Practices for Software Development Teams*.
- Richardson, I. (2002). SPI Models: What Characteristics are Required for Small Software Development Companies? *Software Quality Journal*(14), 101-114.
- Ruiz Carreira, M., & Ramos Roman, I. (2001). Estimación del Coste de la Calidad del Software a través de la Simulación del Proceso de Desarrollo. *Revista Colombiana de Computación*, 2(1), 75-87.
- Schwaber, K. (2004). *Agile Project Management With Scrum*. Redmond, Washington: Microsoft Press.
- Software Engineering Institute (SEI). (2011). *CMMI for SCAMPISM Class A Appraisal Results 2011 Mid-Year Update*. Pittsburgh: Carnegie Mellon.
- Software Engineering Institute (SEI). (2013). *Maturity Profile Reports*. Pittsburgh: CMMI Institute.
- Sommerville, I. (2005). *Ingeniería Del Software*. Madrid, España: Pearson Educación.
- Sutherland, J. (2012). *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework*. Cambridge: scruminc. Recuperado el 10 de 11 de 2011, de <http://jeffsutherland.com/ScrumPapers.pdf>
- Sutherland, J., & Schwaber, K. (2011). *The Definitive Guide of Scrum: The Rules of the Game*. Scrum.Org.
- Sutherland, J., Ruseng Jakobsen, C., & Johnson, K. (2005). *Scrum and CMMI Level 5: The Magic Potion for Code Warriors*. Denmark. Recuperado el 04 de 06 de 2013
- Trudel, S., Lavoie, J. M., Paré, M. C., & Suryn, W. (2006). PEM: The small company-dedicated software process quality evaluation method combining CMMI and ISO/IEC 14598. *Software Qual J*, 14, 7-23.
- Trudel, S., Lavoie, J.-M., Paré, M.-C., & Suryn, W. (2006). PEM: The small company-dedicated software process quality evaluation method combining CMMI and ISO/IEC 14598. *Software Quality Journal*, 7-23.
- Van Soligen, R., & Berghout, E. (1999). *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. Londres: McGraw-Hill.
- Version ONE. (2013). *7th ANNUAL STATE of AGILE VERSIONONE® Agile Made Easier DEVELOPMENT*. Recuperado el 25 de 09 de 2013, de Version ONE: <http://www.versionone.com/pdf/7th-Annual-State-of-Agile-Development-Survey.pdf>
- Wikipedia. (s.f.). *Framework*. Recuperado el 02 de 12 de 2011, de <http://es.wikipedia.org/wiki/Framework>
- Yang, W., Shen, M., Su, H., Rong, G., & Shao, D. (2011). Integrating PSP with agile process: a systematic review. En H. Zhang, Y. Yang, & R. Conradi, *EAST 2011 - Proceeding of the 1st International Workshop on Evidential Assessment of Software Technologies, In conjunction with ENASE 2011* (págs. 36-43). Beijing.

ANEXOS

Anexo 1. Tabla de Registro de Estimación de tiempo vs Real por Tarea

ID Tarea	Tiempo Estimado (Hrs)	Tiempo Real (Hrs)	Diferencia
RELOJ-10	0.50	0.00	0.50
INVENT-447	3.00	3.13	0.13
ICUATRO-601	1.50	0.00	1.50
ICUATRO-599	3.00	2.93	0.07
ICUATRO-598	0.00	0.00	0.00
ICUATRO-596	6.00	2.52	3.48
ICUATRO-595	2.00	5.23	3.23
ICUATRO-594	1.00	6.38	5.38
ICUATRO-593	6.00	0.25	5.75
ICUATRO-592	0.42	0.15	0.27
ICUATRO-591	0.50	1.42	0.92
ICUATRO-590	1.00	1.05	0.05
ICUATRO-589	3.00	2.00	1.00
ICUATRO-588	2.00	1.83	0.17
ICUATRO-585	12.33	13.10	0.77
ICUATRO-583	1.00	1.57	0.57
ICUATRO-582	0.50	0.33	0.17
ICUATRO-581	4.00	0.00	4.00
ICUATRO-580	1.00	0.00	1.00
ICUATRO-579	1.00	0.00	1.00
ICUATRO-578	0.67	0.00	0.67
ICUATRO-577	0.67	0.00	0.67
ICUATRO-576	0.50	0.00	0.50
ICUATRO-575	1.50	0.00	1.50
ICUATRO-574	0.50	0.00	0.50
ICUATRO-573	0.50	0.00	0.50
ICUATRO-572	0.50	0.00	0.50
ICUATRO-571	0.33	0.00	0.33
ICUATRO-570	0.33	0.00	0.33
ICUATRO-569	2.00	0.00	2.00
ICUATRO-568	2.00	2.03	0.03
ICUATRO-567	0.50	0.00	0.50
ICUATRO-566	0.50	0.00	0.50
ICUATRO-565	0.67	0.00	0.67
ICUATRO-564	4.50	0.00	4.50
ICUATRO-563	0.33	0.50	0.17

ICUATRO-562	1.50	0.98	0.52
ERPWEB-47	8.00	10.25	2.25
EDUWIN-104	3.00	9.67	6.67
EDUWIN-101	0.00	0.18	0.18
EDUWIN-99	5.00	0.00	5.00
EDUWEB-17	2.00	0.93	1.07
EDUWEB-16	4.00	0.95	3.05
SAG-2	17.00	11.23	5.77
SAG-1	5.00	6.85	1.85
INVENT-455	1.17	0.58	0.58
INVENT-454	3.00	1.65	1.35
INVENT-453	2.00	2.03	0.03
ICUATRO-666	6.00	18.12	12.12
ICUATRO-665	2.00	1.22	0.78
ICUATRO-664	0.50	1.68	1.18
ICUATRO-652	1.50	0.70	0.80
ICUATRO-651	28.00	16.88	11.12
ICUATRO-646	4.00	1.20	2.80
ICUATRO-645	0.67	0.50	0.17
ICUATRO-644	4.67	0.00	4.67
ERPWEB-49	8.00	43.12	35.12
EDUWIN-110	4.00	0.58	3.42
CONTA-134	2.00	1.83	0.17
EDUWIN-78	8.00	11.75	3.75
ICUATRO-791	8.00	0.00	8.00
ICUATRO-790	4.00	10.98	6.98
ICUATRO-789	4.00	0.00	4.00
ICUATRO-788	3.00	22.70	19.70
ICUATRO-787	3.00	0.00	3.00
ICUATRO-786	4.00	7.30	3.30
ICUATRO-785	0.00	0.00	0.00
ICUATRO-773	5.00	12.08	7.08
ICUATRO-759	0.00	0.00	0.00
ICUATRO-715	1.00	0.65	0.35
ICUATRO-714	0.75	1.03	0.28
ICUATRO-713	8.00	3.38	4.62
ICUATRO-712	3.50	8.27	4.77
ICUATRO-707	2.00	12.10	10.10
ICUATRO-703	2.00	2.35	0.35
ICUATRO-702	5.00	16.50	11.50
ICUATRO-701	4.00	5.97	1.97

ICUATRO-763	1.00	0.00	1.00
ICUATRO-777	8.00	7.92	0.08
ICUATRO-776	4.00	4.02	0.02
ICUATRO-657	3.00	0.82	2.18
SAG-3	4.00	8.08	4.08
ICUATRO-916	3.00	2.57	0.43
ICUATRO-914	4.00	1.90	2.10
ICUATRO-913	5.00	5.07	0.07
ICUATRO-911	0.50	2.12	1.62
ICUATRO-910	1.50	2.02	0.52
ICUATRO-908	8.00	10.40	2.40
ICUATRO-907	8.00	15.82	7.82
ICUATRO-906	20.00	9.17	10.83
ICUATRO-900	0.50	0.57	0.07
ICUATRO-899	16.00	13.07	2.93
ICUATRO-895	2.00	6.67	4.67
ICUATRO-894	5.00	4.47	0.53
ICUATRO-886	24.00	9.68	14.32
ICUATRO-885	8.00	11.03	3.03
ICUATRO-884	40.00	0.00	40.00
ICUATRO-880	7.00	0.00	7.00
ICUATRO-888	8.00	15.95	7.95
ICUATRO-1046	6.00	3.43	2.57
ICUATRO-1044	0.00	0.50	0.50
ICUATRO-1032	6.00	16.00	10.00
ICUATRO-1029	4.00	14.67	10.67
ICUATRO-1022	6.00	0.00	6.00
ICUATRO-1021	4.00	12.95	8.95
ICUATRO-1020	12.00	0.00	12.00
ICUATRO-1019	8.00	0.00	8.00
ICUATRO-1014	8.00	1.40	6.60
ICUATRO-1013	8.00	0.00	8.00
ICUATRO-1012	0.00	1.57	1.57
ICUATRO-1011	0.00	5.67	5.67
ICUATRO-1010	0.00	4.23	4.23
ICUATRO-1008	8.00	7.92	0.08
ICUATRO-989	4.00	0.00	4.00
ICUATRO-988	3.00	0.00	3.00
ICUATRO-882	8.00	0.00	8.00
ICUATRO-1036	8.00	15.42	7.42
ICUATRO-1018	8.00	5.90	2.10

ICUATRO-1015	6.00	1.20	4.80
EDUWIN-133	12.00	0.00	12.00
SNCERP-16	1.50	0.00	1.50
SNCERP-15	8.00	7.27	0.73
ICUATRO-1098	16.00	2.17	13.83
ICUATRO-1097	16.00	9.42	6.58
ICUATRO-1093	5.00	4.05	0.95
ICUATRO-994	6.00	4.02	1.98
ICUATRO-887	12.00	18.10	6.10
ICUATRO-1101	0.00	11.77	11.77
ICUATRO-1100	0.00	10.27	10.27
ICUATRO-896	20.00	2.48	17.52
ICUATRO-1195	0.67	3.07	2.40
ICUATRO-1193	8.00	13.57	5.57
ICUATRO-1192	16.00	10.42	5.58
ICUATRO-1191	2.00	1.20	0.80
ICUATRO-1190	3.50	3.52	0.02
ICUATRO-1186	9.50	16.52	7.02
ICUATRO-1185	16.00	13.08	2.92
ICUATRO-1184	2.50	1.93	0.57
ICUATRO-1183	7.50	14.08	6.58
ICUATRO-1182	5.67	17.83	12.17
ICUATRO-1143	3.50	1.88	1.62
ICUATRO-1141	6.00	0.00	6.00
ICUATRO-1140	8.00	10.08	2.08
ICUATRO-1138	4.00	9.05	5.05
ICUATRO-1187	9.00	4.80	4.20
ICUATRO-1524	12.00	22.65	10.65
SWIN-2	4.00	0.00	4.00
SWBI-2	12.00	12.50	0.50
NOMINA-146	0.00	0.00	0.00
ICUATRO-1488	1.00	0.00	1.00
ICUATRO-1486	8.00	0.00	8.00
ICUATRO-1480	12.00	0.00	12.00
ICUATRO-1465	18.00	8.62	9.38
ICUATRO-1464	12.00	0.00	12.00
ICUATRO-1459	3.00	0.00	3.00
ICUATRO-1409	3.00	0.00	3.00
EDUWIN-146	16.00	1.43	14.57
CONTA-146	0.00	0.00	0.00
ICUATRO-1487	5.00	4.57	0.43

ERPWEB-56	40.00	43.52	3.52
ICUATRO-1574	2.00	0.00	2.00
ICUATRO-1573	2.00	6.02	4.02
ICUATRO-1571	24.00	13.82	10.18
ICUATRO-1570	16.00	13.70	2.30
ICUATRO-1569	16.00	14.17	1.83
ICUATRO-1567	3.00	1.05	1.95
ICUATRO-1566	0.00	0.00	0.00
ICUATRO-1565	0.00	0.00	0.00
ICUATRO-1564	0.00	0.00	0.00
ICUATRO-1563	0.00	0.00	0.00
ICUATRO-1562	0.00	0.00	0.00
ICUATRO-1561	0.00	0.00	0.00
ICUATRO-1557	4.00	5.28	1.28
ICUATRO-1556	9.00	4.35	4.65
ICUATRO-1536	3.00	3.07	0.07
ICUATRO-1535	4.00	1.40	2.60
ICUATRO-1502	8.00	10.37	2.37
ICUATRO-1497	12.00	29.77	17.77
ICUATRO-1572	24.00	34.05	10.05
ICUATRO-1718	0.00	0.00	0.00
ICUATRO-1717	0.00	0.00	0.00
ICUATRO-1716	0.00	0.00	0.00
ICUATRO-1713	0.00	0.00	0.00
ICUATRO-1712	0.00	0.00	0.00
ICUATRO-1711	0.00	0.00	0.00
ICUATRO-1710	0.00	0.00	0.00
ICUATRO-1705	0.00	0.00	0.00
ICUATRO-1704	0.00	0.00	0.00
ICUATRO-1703	1.00	5.38	4.38
ICUATRO-1702	4.00	3.77	0.23
ICUATRO-1699	2.00	0.00	2.00
ICUATRO-1696	0.50	4.22	3.72
ICUATRO-1692	1.50	0.00	1.50
ICUATRO-1689	1.00	2.87	1.87
ICUATRO-1688	1.50	4.83	3.33
ICUATRO-1687	1.00	9.37	8.37
ICUATRO-1683	8.00	0.03	7.97
ICUATRO-1682	8.00	16.33	8.33
ICUATRO-1677	2.00	0.00	2.00
ICUATRO-1644	4.00	18.30	14.30

ICUATRO-1684	12.00	14.35	2.35
ICUATRO-1802	2.00	0.30	1.70
ICUATRO-1798	0.00	0.00	0.00
ICUATRO-1797	0.00	0.00	0.00
ICUATRO-1796	0.00	0.00	0.00
ICUATRO-1795	0.00	0.00	0.00
ICUATRO-1793	8.00	4.15	3.85
ICUATRO-1785	1.50	0.57	0.93
ICUATRO-1783	3.00	1.25	1.75
Total	1063.83	1019.42	819.52

Anexo 2. Resultados preguntas encuesta de evaluación del producto

1. ¿Hace cuanto hace uso de nuestros productos?

Opciones de respuesta–	Respuestas–
Menos de un mes	6,12% 3
Entre uno y seis meses	4,08% 2
Entre seis meses y un año	14,29% 7
Entre uno y dos años	40,82% 20
Más de tres años	34,69% 17
Total	49

2. Seleccione las aplicaciones que su organización maneja con la organización

Opciones de respuesta–	Respuestas–
Business Management (ERP)	88,10% 37
CRM	4,76% 2
Software de Universidades	0% 0
Software de Colegios	7,14% 3
Total de encuestados: 42	

3. Responda esta respuesta si selecciono Business Management (ERP) en la respuesta anterior, ¿Cuáles procesos de la aplicación tiene implementados en su organización?

Opciones de respuesta–	Respuestas–
Nomina	60% 24
Inventarios	70% 28
Gestión Financiera	70% 28
Gestión Comercial	60% 24
Gestión de Calidad	5% 2

Control de Asistencia	5% 2
Producción	35% 14
Control de Mantenimiento	5% 2
Total de encuestados: 40	

4. En general, ¿considera que las aplicaciones de la organización cumplen con las necesidades de su organización?

Opciones de respuesta–	Respuestas–
Completamente	39,58% 19
Parcialmente	60,42% 29
Muy Poco	0% 0
Nada	0% 0
Total	48

5. En general como califica la calidad de las aplicaciones que su organización maneja con la organización.

Opciones de respuesta–	Respuestas–
Excelente	19,57% 9
Buena	67,39% 31
Regular	13,04% 6
Deficiente	0% 0
Total	46

6. En general, ¿Cada cuánto recibe actualizaciones de las aplicaciones de la organización?

Opciones de respuesta–	Respuestas–
Cada semana	0% 0

Opciones de respuesta–	Respuestas–
Cada 15 días	17,07% 7
Cada Mes	24,39% 10
Cada Seis Meses	9,76% 4
Cada Año	29,27% 12
Nunca	19,51% 8
Total	41

7. Cuando recibe actualizaciones de las aplicaciones de la organización, por lo general la actualización:

Opciones de respuesta–	Respuestas–
Tiene mejoras y no hay errores	9,76% 4
Tiene mejoras y pocos errores	56,10% 23
Tiene mejoras y muchos errores	24,39% 10
No tiene mejoras y pocos errores	7,32% 3
No tiene mejoras y muchos errores	2,44% 1
Total	41

8. Cuando reporta un error de las aplicaciones de la organización, ¿cuánto debe esperar para que se corrija este error?

Opciones de respuesta–	Respuestas–
Menos de una semana	36,59% 15
Entre una semana y 15 días	36,59% 15
Un mes	17,07% 7
Seis meses	4,88% 2
Nunca	4,88% 2

Opciones de respuesta–	Respuestas–
Total	41

9. Cuando recibe la corrección a un error que reportó de las aplicaciones de la organización, por lo general:

Opciones de respuesta–	Respuestas–
Se corrige el error reportado	46,34% 19
Se corrige el error pero aparece un nuevo error	53,66% 22
No se corrige el error	0% 0
No se corrige el error y aparece un nuevo error	0% 0
Total	41

10 ¿Recomendaría las aplicaciones de la organización a otras organizaciones?

Opciones de respuesta–	Respuestas–
Si	80% 32
No	20% 8
Total	40

Anexo 3. Resultados detallados encuesta de equipo de desarrollo

1. ¿ Considera que actualmente existe un proceso definido de desarrollo de software dentro de la organización?

Opciones de respuesta	Respuestas
Totalmente	33,33% 2
Parcialmente	66,67% 4
No está definido	0% 0
Total	6

2. Considera que posee la suficiente capacitación de la aplicación del marco de trabajo de SCRUM dentro de la organización?

Opciones de respuesta	Respuestas
Totalmente	66,67% 4
Parcialmente	33,33% 2
No tengo conocimiento de SCRUM	0% 0
Total	6

3. Considera que posee la suficiente capacitación de la aplicación de PSP dentro de su proceso de desarrollo de software?

Opciones de respuesta	Respuestas
Totalmente	50% 3
Parcialmente	50% 3
No tengo conocimiento de PSP	0% 0
Total	6

4. Considera que el marco de trabajo SCRUM y el modelo de proceso PSP están integrados dentro de un proceso definido de desarrollo dentro de la organización?

Opciones de respuesta	Respuestas
Se encuentran integrados en su totalidad	66,67% 4
Se encuentran integrados parcialmente	16,67% 1
Se manejan de manera independiente sin integración	16,67% 1

Opciones de respuesta	Respuestas
Total	6

5. Considera que el proceso de desarrollo esta guiado por un modelo de control de calidad estándar definido?

Opciones de respuesta	Respuestas
Si	50% 3
No	50% 3
Total	6

6. Considera que existe la suficiente documentación para la ejecución de las actividades de desarrollo asignadas?

Opciones de respuesta	Respuestas
Totalmente	16,67% 1
Parcialmente	83,33% 5
No se ha definido documentación	0% 0
No es necesaria la documentación	0% 0
Total	6

7. Considera que la estimación de las tareas de desarrollo dentro de los proyectos son:

Opciones de respuesta	Respuestas
Completamente acertadas	0% 0
Poco acertadas	100% 6
Nada acercadas a la realidad	0% 0
Total	6

8. Considera la aplicación de SCRUM y PSP dentro del proceso de desarrollo, favorece al ambiente de trabajo del equipo?

Opciones de respuesta	Respuestas
Totalmente	100% 6
Puede mejorar	0% 0

Opciones de respuesta	Respuestas
De ninguna manera	0% 0
Total	6

9.Cuál es la percepción de calidad que posee del productos desarrollados por la organización?

Opciones de respuesta	Respuestas
Excelente	16,67% 1
Buena	66,67% 4
Aceptable	16,67% 1
Deficiente	0% 0
Total	6

10. Que recomendaciones puede dar para mejorar el proceso de desarrollo que actualmente se maneja en la organización?

-Mejorar en cuanto a documentación.
Lo más importante son las pruebas externas y que exista una retroalimentación
Sabemos que estimar el tiempo para el desarrollo es importante, pero debido a factores externos siempre es difícil cumplir con los tiempos estimados.
Que se pueda dedicar mas tiempo a la implementacion de los items de una metodologia de desarrollo, y que no se exija terminar todas las tareas en un tiempo muy corto, porque esto no deja realizar bien la practica de SCRUM y PSP
Formalizar el proceso de desarrollo e integrarlo, de igual forma establecer dentro del proceso pruebas, y un control de calidad

GUIONES PSP 0

SCRUM-PSP0 Guión del proceso

Propósito	Guía el desarrollo de un módulo o nivel de un programa	
Criterio de entrada	Requerimiento del sprint <i>backlog</i> a desarrollar Resumen del plan del proyecto SCRUM-PSP0.1 <i>Logs</i> para registro de tiempo y defectos Estándares de codificación y lista de tipos de defectos Cronometro (Opcional)	
Paso	Actividades	Descripción
1	Planeación	Completar el plan de resumen del proyecto Diligenciar el tiempo en el log de registro de tiempos.
2	Desarrollo	Diseñe el programa Implemente el diseño Compile el programa, repare y registre cualquier defecto encontrado Pruebe el programa, repare y registre cualquier defecto encontrado Complete el tiempo en el log de registro de tiempos.
3	Postmortem	Complete el Plan de resumen del proyecto con el tiempo actual y defectos
Criterios de salida	Un programa probado Información del resumen de plan del proyecto completa con los datos de estimación y datos actuales <i>Logs</i> de registros de tiempos y defectos completos	

SCRUM-PSP0 Guión de planeación

Propósito	Guiar el proceso de planeación.	
Criterios de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Formulario de Resumen de plan del proyecto Log de registros de tiempo.	
Paso	Actividades	Descripción
1	Estimación de recursos	Realice su mejor estimación del tiempo que requiere para desarrollar el requerimiento dado (Puede ser igual o variar a la que ya se realizó en el <i>sprint planning</i>)
Criterios de salida		Resumen de plan de proyecto con el tiempo de estimación. Diligenciar el tiempo en el log de registros de tiempos.

SCRUM-PSP0 Guión de Desarrollo

Purpose	Guía para el desarrollo de programas pequeños.	
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Resumen de plan de proyecto con la información de tiempos. <i>Logs</i> de registro de tiempos y defectos Lista de tipos de defectos y estándares de codificación. Plantilla de diseño	
Paso	Actividades	Descripción
1	Diseño	Revise los requerimientos y produzca un diseño que le permita llegar a ellos usando las guías establecidas. Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registro de tiempos.
2	Codificación	Implemente el diseño Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registros de tiempos
3	Pruebas	Pruebe hasta que todas las pruebas no arrojen ningún error. Repáre todos los defectos encontrados Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registros de tiempos
Criterios de salida		A la salida se tendrá un programa probado <i>Logs</i> de tiempos y defectos completos.

PSP0 Postmortem Guión

Propósito	Guía para el proceso de Postmortem de PSP	
Criterios de entrada	Requerimiento del <i>sprint backlog</i> desarrollado, <i>sprint backlog</i> Resumen del plan de proyecto con la información de tiempos <i>Logs</i> de tiempos y defectos completos Un programa probado y corriendo conforme a los estándares de codificación	
Paso	Actividades	Descripción
1	Registro de defectos	Revisar el Resumen del plan del proyecto para verificar que todos los defectos se encuentren registrados en todas las fases. Usando su mejor estimación, registre cualquier defecto omitido.
2	Revisión de inconsistencia de datos en defectos	Chequee que los datos de cada defecto en el log de registro de defectos se encuentre exacto y completo Verifique que el número de defectos inyectados y removidos por fase sean razonables y correctos Usando su mejor estimación, corrija cualquier pérdida o datos de defectos incorrectos.
3	Tiempo	Revisar en el log de registro de tiempos errores u omisiones. Use su mejor estimación, y corrija cualquier dato de tiempo perdido o incompleto.
Criterios de salida	Un programa probado a fondo Completar el Resumen del plan del proyecto <i>Logs</i> de registros de tiempos y defectos completos.	

Resumen del Plan de Proyecto PSP 0.1

Desarrollador	_____	Fecha	_____
Incidencia	_____	Programa#	_____

Tiempos en Fase (min.)	<i>Plan</i>	Actual	A la Fecha	A la Fecha %
Planeación		_____	_____	_____
Diseño		_____	_____	_____
Codificación		_____	_____	_____
Pruebas		_____	_____	_____
Postmortem		_____	_____	_____
Total	_____	_____	_____	_____

Defectos Inyectados	Actual	A la Fecha	A la Fecha %
Planeación	_____	_____	_____
Diseño	_____	_____	_____
Codificación	_____	_____	_____
Pruebas	_____	_____	_____
Total Desarrollo	_____	_____	_____

Defectos removidos	Actual	A la Fecha	A la Fecha %
Planeación	_____	_____	_____
Diseño	_____	_____	_____
Codificación	_____	_____	_____
Pruebas	_____	_____	_____
Total desarrollo	_____	_____	_____
Después de desarrollo	_____	_____	_____

Resumen del Plan de Proyecto PSP 0

Propósito	Donde se deposita la información del plan y los datos actuales de los programas o partes de el
General	A la fecha es el total de valores actuales que se lleva a la fecha de todos los productos desarrollados.
Cabecera	Introduzca su nombre y fecha. Introduzca el ID de la incidencia y numero de programa.
Tiempo por Fase	Introduzca el total estimado de tiempo de desarrollo. Introduzca el tiempo actual por fase y el tiempo total. A la fecha: Introduzca la suma de los tiempos actuales para este programa más los tiempos a la fecha del programa más reciente desarrollado. % A la fecha: Introduzca el porcentaje de cada fase a la fecha.
Defectos Inyectados	Introduzca el número de defectos actuales por fase y el total de defectos actuales. A la fecha: Introduzca la suma de los defectos actuales inyectados por cada fase y los valores a la fecha del programa más reciente hecho previamente. % A la fecha: Introduzca el porcentaje a la fecha de defectos inyectados en cada fase.
Defectos Removidos	A la fecha: Introduzca los defectos actuales removidos por fase más los valores a la fecha del programa más reciente hecho previamente. % A la fecha: Introduzca el porcentaje de los defectos a la fecha removidos por fase. Después de desarrollo: Registre cualquier defecto encontrado posteriormente durante pruebas, uso, reusó o modificación.

SCRUM-PSP0.1 Guión del proceso

Propósito	Guía el desarrollo de un módulo o nivel de un programa	
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Resumen del plan del proyecto SCRUM-PSP0.1 <i>Logs</i> para registro de tiempo y defectos Estándares de codificación y lista de tipos de defectos Cronometro (Opcional)	
Paso	Actividades	Descripción
1	Planeación	Completar el plan de resumen del proyecto Diligenciar el tiempo en el log de registro de tiempos.
2	Desarrollo	Diseñe el programa Implemente el diseño Compile el programa, repare y registre cualquier defecto encontrado pruebe el programa, repare y registre cualquier defecto encontrado Complete el tiempo en el log de registro de tiempos.
3	Postmortem	Complete el Plan de resumen del proyecto con el tiempo actual, defectos y tamaño del programa.
Criterios de salida	Un programa probado Información del resumen de plan del proyecto completa con los datos de estimación y datos actuales Registro PIP completo <i>Logs</i> de registros de tiempos y defectos completos	

SCRUM-PSP0.1 Guión de planeación

Propósito	Guiar el proceso de planeación.	
Criterios de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Formulario de Resumen de plan del proyecto Log de registros de tiempo.	
Paso	Actividades	Descripción
1	<i>Estimación de tamaño</i>	<i>Haga su mejor estimación del código a adicionar y modificar. Ingrese el plan de tamaño a estimado en el Resumen de plan del proyecto.</i>
2	Estimación de recursos	Realice su mejor estimación del tiempo que requiere para desarrollar el requerimiento dado (Puede ser igual o variar a la que ya se realizó en el <i>sprint planning</i>) <i>Use el campo "% a la fecha" del más reciente programa desarrollado como guía para distribuir el tiempo de desarrollo en las diferentes fases.</i>
Criterios de salida	Resumen de plan de proyecto con los tiempos de desarrollo distribuidos. Diligenciar el tiempo en el log de registros de tiempos.	

SCRUM-PSP0.1 Guión de Desarrollo

Propósito	Guía para el desarrollo de programas pequeños.	
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Resumen de plan de proyecto con la información de tiempos. <i>Logs</i> de registro de tiempos y defectos Lista de tipos de defectos y estándares de codificación. Plantilla de diseño	
Paso	Actividades	Descripción
1	Diseño	Revise los requerimientos y produzca un diseño que le permita llegar a ellos siguiendo las guías establecidas Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registro de tiempos.
2	Codificación	Implemente el diseño siguiendo los estándares de codificación. Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registros de tiempos
3	Pruebas	Pruebe hasta que todas las pruebas no arrojen ningún error. Repáre todos los defectos encontrados Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registros de tiempos
Criterios de salida	A la salida se tendrá un programa probado y conforme a los estándares de codificación. <i>Logs</i> de tiempos y defectos completos.	

PSP0.1 Guión Postmortem

Propósito	Guía para el proceso de <i>Postmortem</i> de PSP	
Criterios de entrada	Requerimiento del <i>sprint backlog</i> desarrollado Resumen del plan de proyecto con la información de tiempos <i>Logs</i> de tiempos y defectos completos Un programa probado y corriendo conforme a los estándares de codificación,	
Paso	Actividades	Descripción
1	Registro de defectos	Revisar el Resumen del plan del proyecto para verificar que todos los defectos se encuentren registrados en todas las fases. Usando su mejor estimación, registre cualquier defecto omitido.
2	Revisión de inconsistencia de datos en defectos	Chequee que los datos de cada defecto en el log de registro de defectos se encuentre exacto y completo Verifique que el número de defectos inyectados y removidos por fase sean razonables y correctos Usando su mejor estimación, corrija cualquier pérdida o datos de defectos incorrectos.
3	Tamaño	<i>Cuenta el tamaño del programa (LOC).</i> <i>Determine el tamaño de la base, borrado, modificado, reusado, total y código reusable nuevo (Nota: El tamaño del código adicionado y modificado es calculado)</i> <i>Ingrese los datos en el resumen del plan del proyecto.</i>
4	Tiempo	Revisar en el log de registro de tiempos errores u omisiones. Use su mejor estimación, y corrija cualquier dato de tiempo perdido o incompleto.
Criterios de salida	Un programa probado a fondo de acuerdo a los estándares de codificación. Completar el Resumen del plan del proyecto <i>Completar el PSP describiendo los problemas, sugerencias de mejora, y lecciones aprendidas</i> <i>Logs</i> de registros de tiempos y defectos completos.	

Resumen del Plan de Proyecto PSP 0.1

Desarrollador _____ Fecha _____
 Incidencia _____ Programa# _____

Tamaño del programa	Actual	A la fecha
Base (B)	_____	
	(Medida)	
Borradas (D)	_____	
	(Contadas)	
Modificadas (M)	_____	
	(Contadas)	
Agregadas (A)	_____	
	($T - B + D - R$)	
Reusadas (R)	_____	
	(Contadas)	
Agregadas y Modificadas (A+M)	_____	
	($A + M$)	
Tamaño Total (T)	_____	
	(Medidas)	
Total nuevas reusables		

Tiempos en Fase (min.)	Plan	Actual	A la Fecha	A la Fecha %
Planeación	_____	_____	_____	_____
Diseño	_____	_____	_____	_____
Codificación	_____	_____	_____	_____
Pruebas	_____	_____	_____	_____
Postmortem	_____	_____	_____	_____
Total	_____	_____	_____	_____

Defectos Inyectados	Actual	A la Fecha	A la Fecha %
Planeación	_____	_____	_____
Diseño	_____	_____	_____
Codificación	_____	_____	_____
Pruebas	_____	_____	_____
Total Desarrollo	_____	_____	_____

Defectos removidos	Actual	A la Fecha	A la Fecha %
Planeación	_____	_____	_____
Diseño	_____	_____	_____
Codificación	_____	_____	_____
Pruebas	_____	_____	_____
Total desarrollo	_____	_____	_____
Después de desarrollo	_____	_____	_____

Resumen del Plan de Proyecto PSP 0.1

Propósito	Donde se depositara la información del plan y los datos actuales de los programas o partes de el
General	<i>Usar la más apropiada medida de tamaño, como LOC o contador de elementos.</i> A la fecha es el total de valores actuales que se lleva a la fecha de todos los productos desarrollados.
Cabecera	Introduzca su nombre y fecha. Introduzca el ID de la incidencia y numero de programa.
Tiempo por Fase	Introduzca el total estimado de tiempo de desarrollo. <i>Distribuya el estimado de tiempo total entre las fases de desarrollo de acuerdo al porcentaje a la fecha del programa desarrollado más recientemente.</i> Introduzca el tiempo actual por fase y el tiempo total. A la fecha: Introduzca la suma de los tiempos actuales para este programa más los tiempos a la fecha del programa más reciente desarrollado. % A la fecha: Introduzca el porcentaje de cada fase a la fecha.
Defectos Inyectados	Introduzca el número de defectos actuales por fase y el total de defectos actuales. A la fecha: Introduzca la suma de los defectos actuales inyectados por cada fase y los valores a la fecha del programa más reciente hecho previamente. % A la fecha: Introduzca el porcentaje a la fecha de defectos inyectados en cada fase.
Defectos Removidos	A la fecha: Introduzca los defectos actuales removidos por fase más los valores a la fecha del programa más reciente hecho previamente. % A la fecha: Introduzca el porcentaje de los defectos a la fecha removidos por fase. Después de desarrollo: Registre cualquier defecto encontrado posteriormente durante pruebas, uso, reusó o modificación.

Propuesta de mejora del proceso PSP (PIP)

Desarrollador

Incidencia

Fecha

Programa#

Descripción del problema

Brevemente describa los problemas que usted encontró.

Descripción del propósito

Brevemente describa las mejoras al proceso que usted propone

Otras notas y comentarios

Escriba cualquier otro comentario u observación que describa su experiencia o ideas de mejora.

Instrucciones Propuesta de mejora del proceso PSP (PIP)

Propósito	<p>Provee una guía para registrar los problemas del proceso e ideas de mejora .</p> <p>Provee un registro ordenado de su proceso de sus ideas de mejora.</p> <p>Para registrar cualquier otra nota u observación.</p>
General	<p>Use el formulario PIP para</p> <p>Registre ideas de mejora del proceso que se le puedan ocurrir.</p> <p>Establezca prioridades para sus planes de mejora.</p> <p>Describa las lecciones aprendidas y condiciones inusuales.</p> <p>Mantenga los formularios PIP a la mano mientras use PSP.</p> <p>Registre problemas durante el proceso aun sin tener propuesta de soluciones</p> <p>Presente el informe PIP en cada proyecto PSP</p>
Cabecera	<p>Introduzca su nombre y fecha.</p> <p>Introduzca el ID de la incidencia y numero de programa...</p>
Descripción del problema	Brevemente describa cualquier problema o experiencia que lo llevaron al PIP.
Descripción de la propuesta	Describa de la manera más explícita las propuestas de mejora
Otras notas y comentarios	<p>Brevemente describe cualquier otra observación o hecho que le pueda ayudar a usted.</p> <p>Recuerde que hizo mientras escribió el programa.</p> <p>Recuerde cualquier idea para la mejora futura.</p>

GUINES PSP 1

SCRUM-PSP1 Guión del proceso

Propósito	Guía el desarrollo de un modulo o nivel de un programa	
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Resumen del plan del proyecto SCRUM-PSP0.1 <i>Estimación por talla del requerimiento</i> <i>Datos de tamaño y tiempo históricos (Estimados y actuales)</i> <i>Logs</i> de registros de tiempos y defectos Estándares de codificación y lista de tipos de defectos Cronometro (Opcional)	
Paso	Actividades	Descripción
1	Planeación	Obtenga el requerimiento del <i>sprint backlog</i> a desarrollar <i>Use el método PROBE para</i> estimar el código a adicionar y modificar en este programa <i>Complete la plantilla de estimación de tamaño</i> <i>Utilice el método PROBE</i> para estimar el tiempo requerido de desarrollo. Completar el plan de resumen del proyecto Diligenciar el tiempo en el log de registro de tiempos.
2	Desarrollo	Diseñe el programa Implemente el diseño Compile el programa, repare y registre cualquier defecto encontrado pruebe el programa, repare y registre cualquier defecto encontrado Complete el tiempo en el log de registro de tiempos.
3	Postmortem	Complete el Plan de resumen del proyecto con el tiempo actual, defectos y tamaño del programa
Criterios de Salida	Un programa probado Información del resumen de plan del proyecto completa con los datos de estimación y datos actuales <i>Plantilla de estimación de tamaño completa</i> <i>Reporte de pruebas completo</i> Registro PIP completo <i>Logs</i> de registros de tiempos y defectos completos	

SCRUM-PSP1 Guión de planeación

Propósito	Guiar el proceso de planeación .	
Criterios de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Formulario de Resumen de plan del proyecto <i>Plantilla de estimación de tamaño</i> <i>Datos de tamaño y tiempo históricos (Estimados y actuales)</i> Log de registros de tiempo.	
Paso	Actividades	Descripción
1	<i>Estimación de tamaño</i>	<i>Produzca un diseño conceptual del programa</i> <i>Utilice el método PROBE</i> para estimar el código a adicionar y modificar en este programa. <i>Complete la plantilla de estimación de tamaño y el Resumen de plan del proyecto.</i>
2	Estimación de recursos	<i>Use el método PROBE para</i> estimar el tiempo requerido para desarrollar este programa. Use el campo "% a la fecha" del más reciente programa desarrollado como guía para distribuir el tiempo de desarrollo en las diferentes fases.
Criterios de salida	<i>Diseño conceptual del programa</i> <i>Plantilla de estimación de tamaño completa</i> Resumen de plan de proyecto con los tiempos de desarrollo distribuidos. Diligenciar el tiempo en el log de registros de tiempos.	

SCRUM-PSP0.1 Guión de Desarrollo

Propósito	Guía para el desarrollo de programas pequeños.	
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Resumen de plan de proyecto con la información de tiempos. <i>Logs</i> de registro de tiempos y defectos Lista de tipos de defectos y estándares de codificación. Plantilla de diseño	
Paso	Actividades	Descripción
1	Diseño	Revise los requerimientos y produzca un diseño que le permita llegar a él siguiendo las guías establecidas. Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registro de tiempos.
2	Codificación	Implemente el diseño siguiendo los estándares de codificación Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registros de tiempos
3	Pruebas	Pruebe hasta que todos las pruebas no tengan ningun error. Repáre todos los defectos encontrados

		<p>Registre en el log de registro de defectos, cualquier defecto encontrado.</p> <p>Registre el tiempo en el log de registros de tiempos</p> <p><i>Completar la plantilla de reporte de pruebas con los test ejecutados y los resultados obtenidos.</i></p>
Criterios de salida		<p>A la salida se tendrá un programa probado y conforme a los estándares de codificación.</p> <p><i>Plantilla de reporte de pruebas completa.</i></p> <p><i>Logs de tiempos y defectos completos.</i></p>

PSP1 Guión Postmortem Guión

Propósito	Guía para el proceso de <i>Postmortem</i> de PSP
Criterios de entrada	Requerimiento del <i>sprint backlog</i> desarrollado Resumen del plan de proyecto con la información de tiempos <i>Plantilla de reportes de pruebas completa.</i> <i>Logs</i> de tiempos y defectos completos Un programa probado y corriendo conforme a los estándares de codificación,

Paso	Actividades	Descripción
1	Registro de defectos	Revisar el Resumen del plan del proyecto para verificar que todos los defectos se encuentren registrados en todas las fases. Usando su mejor estimación, registre cualquier defecto omitido.
2	Revisión de inconsistencia de datos en defectos	Chequee que los datos de cada defecto en el log de registro de defectos se encuentre exacto y completo Verifique que el numero de defectos inyectados y removidos por fase sean razonables y correctos Usando se mejor estimación, corrija cualquier pérdida o datos de defectos incorrectos.
3	Tamaño	Cuenta el tamaño del programa (LOC). <i>Determine el tamaño del código base, borrado, modificado, adiciones a la base, reusado, nuevo reusable, y partes agregadas.</i> <i>Ingrese los datos en la plantilla de estimación de tiempo</i> <i>Determine el total de tamaño del programa</i> <i>Ingrese los datos en el Resumen de plan del proyecto.</i>
4	Tiempo	Revisar en el log de registro de tiempos errores o omisiones. Use su mejor estimación, y corrija cualquier dato de tiempo perdido o incompleto.

Criterios de Salida	Un programa probado a fondo de acuerdo a los estándares de codificación. <i>Plantilla de reporte de pruebas completa</i> Completar el Resumen del plan del proyecto Completar el PSP describiendo los problemas, sugerencias de mejora, y lecciones aprendidas <i>Logs</i> de registros de tiempos y defectos completos.
---------------------	---

Ejemplo PSP1 Resumen del plan de proyecto

Desarrollador _____ Fecha _____
 Incidencia _____ Programa# _____

Resumen LOC/Hora	Plan	Actual	A la fecha	
Tamaño del programa Base (B)	Plan	Actual	To Date	
	(Medida)	(Medida)		
Borradas (D)	(Estimada)	(Contadas)		
Modificadas (M)	(Estimada)	(Contadas)		
Agregadas (A)	(A+M - M)	(T - B + D - R)		
Reusadas (R)	(Estimada)	(Contadas)		
Agregadas y Modificadas (A+M)	(Proyectada)	(A + M)		
Tamaño Total (T)	(A+M + B - M - D + R)	(Medidas)		
Total nuevas reusables				
Tamaño del proxy estimado (E)				
Tiempos en Fase (min.)	Plan	Actual	A la Fecha	A la Fecha %
Planeación				
Diseño				
Codificación				
Pruebas				
Postmortem				
Total				
Tiempos en Fase (min.)				
Defectos Inyectados		Actual	A la Fecha	A la Fecha %
Planeación				
Diseño				
Codificación				
Pruebas				
Total Desarrollo				
Defectos removidos		Actual	A la Fecha	A la Fecha %
Planeación				
Diseño				
Codificación				
Pruebas				
Total desarrollo				
Después de desarrollo				

Resumen del Plan de Proyecto PSP 1

Propósito	Donde se depositara la información del plan y los datos actuales de los programas o partes de el
General	<p>Usar la más apropiada medida de tamaño, como LOC o contador de elementos.</p> <p>A la fecha es el total de valores actuales que se lleva a la fecha de todos los productos desarrollados.</p> <p><i>Una parte puede ser un modulo, componente, producto o sistema.</i></p>
Cabecera	<p>Introduzca su nombre y fecha.</p> <p>Introduzca el ID de la incidencia y numero de programa.</p>
Resumen	<i>Ingrese las líneas a adicionar y modificar por hora planeada, actual y a la fecha.</i>
Tamaño del programa	<p><i>Introduzca lo planeado de código base, borrado, modificado, reusado, nuevo reusable, y tamaño total desde la plantilla de estimación de tamaño</i></p> <p>Ingrese el plan el valor de líneas a adicionar y modificar (A+M) <i>de lo proyectado de adicionar y modificar (P) de la plantilla de estimación de tamaño.</i></p> <p><i>Calcular el plan de tamaño a adicionar $A+M-M$.</i></p> <p><i>Ingrese el tamaño del proxy estimado (E) desde la plantilla de estimación.</i></p> <p>Ingrese el valor de tamaño actual del código base, borrado, modificado reusado, total y nuevo reusable. Calcule el tamaño actual agregado como $T-B+D-R$ y el actual agregado y modificado como $A+M$</p> <p>Introduzca A la fecha el tamaño de lo reusado, agregado, modificado, total y nuevo reusable.</p>
Tiempos en la fase	<p>Introduzca <i>el total de tiempo en cada fase</i> desde el total estimado de tiempo de desarrollo <i>en la plantilla de estimación de tiempo.</i></p> <p>Distribuya el estimado de tiempo total entre las fases de desarrollo de acuerdo al porcentaje a la fecha del programa desarrollado más recientemente.</p> <p>Introduzca el tiempo actual por fase y el tiempo total.</p> <p>A la fecha: Introduzca la suma de los tiempos actuales para este programa más los tiempos a la fecha del programa más reciente desarrollado.</p> <p>% A la fecha: Introduzca el porcentaje de cada fase a la fecha.</p>

Defectos Inyectados	<p>Introduzca el numero de defectos actuales por fase y el total de defectos actuales.</p> <p>A la fecha: Introduzca la suma de los defectos actuales inyectados por cada fase y los valores a la fecha de el programa más reciente hecho previamente..</p> <p>% A la fecha: Introduzca el porcentaje a la fecha de defectos inyectados en cada fase.</p>
Defectos Removidos	<p>A la fecha: Introduzca los defectos actuales removidos por fase más los valores a la fecha de el programa mas reciente hecho previamente.</p> <p>% A la fecha: Introduzca el porcentaje de los defectos a la fecha removidos por fase.</p> <p>Después de desarrollo: Registre cualquier defecto encontrado posteriormente durante pruebas, uso, reusó o modificación.</p>

Guión para estimación mediante PROBE

Propósito	Guía para el proceso de estimación de tamaño y tiempo a través del método PROBE.
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Plantilla de estimación y instrucciones Tamaño por ítem para tipos de partes Log de registros de tiempos Información histórica de tamaño y tiempo.
General	Este guión asume que se esta usando información del tamaño añadido y modificado como los tipos de conteo de tamaño para hacer estimaciones de tiempo y tamaño

Paso	Actividades	Descripción
1	Diseño conceptual	Revisar la tarea del <i>sprint backlog</i> seleccionada y realizar un diseño conceptual.
2	Adiciones de partes	Siga las instrucciones de la plantilla de estimación para estimar las adiciones de partes y las nuevas partes reusables.
3	Partes base y partes reusadas	Para el programa base, estimar el tamaño de código base, eliminado, modificado y añadido. Medir y/o estimar el tamaño de las partes a ser reusadas.
4	Procedimiento de estimación del tamaño	Si se tiene un suficiente tamaño del proxy estimado e información actual del tamaño añadido y modificado (3 o más puntos que correlacionen) usar el procedimiento 4A Si no tienes suficiente dato estimado pero tienes suficiente información actual y planeada del tamaño añadido y modificado(tres o mas puntos que correlacionen), usa el procedimiento 4B Si tiene insuficiente información, o esta no correlaciona, use el procedimiento 4C. Si no tiene información histórica, use el procedimiento 4D.
4A	Procedimiento 4A de Estimación del tamaño	Usando el método de regresión lineal, calcule los parámetros β_0 y β_1 de la información del tamaño añadido y modificado actual y del tamaño del proxy estimado Si el valor absoluto de β_0 no es cercano a 0 (menos que 25% del tamaño esperado del nuevo programa), o β_1 no es cercano a 1.0 (entre 0.5 y 2.0), use el procedimiento 4B.
4B	Procedimiento 4B de Estimación del tamaño	Usando el método de regresión lineal, calcule los parámetros β_0 y β_1 de la información actual y planificado del tamaño añadido y modificado. Si el valor absoluto de β_0 no es cercano a 0 (menos que 25% del tamaño esperado del nuevo programa), o β_1 no es cercano a 1.0 (entre 0.5 y 2.0), use el procedimiento 4C.
4C	Procedimiento 4C de Estimación del tamaño	Si se tiene algún dato sobre el tamaño planificado o actual añadido y modificado, configure $\beta_0 = 0$ y $\beta_1 = (\text{Total actual del tamaño añadido y modificado a la fecha} / \text{Total planificado})$

		del tamaño añadido y modificado a la fecha).
4D	Procedimiento de estimación 4D	Si no se tiene información histórica, use su juicio para estimar el tamaño añadido y modificado.

(continua)

PROBE Guión (continuación)

Paso	Actividades	Descripción
5	Procedimiento de Estimación del tiempo	<p>Si se tiene un suficiente información del tamaño del proxy estimado y del tiempo de desarrollo actual (tres o más puntos que correlacionen), usar el procedimiento 5A.</p> <p>Si no tienes suficiente información del tamaño estimado pero tienes suficiente información actual y planeada del tamaño añadido y modificado (tres o más puntos que correlacionen), use el procedimiento 5B.</p> <p>Si tiene insuficiente información, o esta no correlaciona, use el procedimiento 5C.</p> <p>Si no tiene información histórica, use el procedimiento 5D.</p>
5A	Procedimiento 5A de Estimación del tiempo	<p>Usando el método de regresión lineal, calcule los parámetros β_0 y β_1 de la información del actual tiempo total de desarrollo y del tamaño del proxy estimado</p> <p>Si β_0 no es cercano a 0 (sustancialmente más pequeño que el tiempo de desarrollo esperado para el nuevo programa), o β_1 no está dentro del 50% de $1/(\text{productividad histórica})$, use el procedimiento 5B.</p>
5B	Procedimiento 5B de Estimación del tiempo	<p>Usando el método de regresión lineal, calcule los parámetros β_0 y β_1 de la información del actual tiempo total de desarrollo y del tamaño añadido y modificado planeado.</p> <p>Si β_0 no es cercano a 0 (sustancialmente más pequeño que el tiempo de desarrollo esperado para el nuevo programa), o β_1 no está dentro del 50% de $1/(\text{productividad histórica})$, use el procedimiento 5C.</p>
5C	Procedimiento 5C de Estimación del tiempo	<p>Si tiene información del tamaño añadido y modificado estimado y el tiempo de desarrollo actual, coloque $\beta_0 = 0$ y $\beta_1 = (\text{Total del Tiempo de desarrollo actual a la fecha} / \text{Total estimado del tamaño añadido y modificado a la fecha})$.</p> <p>Si tiene información del tamaño añadido y modificado planeado y el tiempo de desarrollo actual, coloque $\beta_0 = 0$ y $\beta_1 = (\text{Total del Tiempo de desarrollo actual a la fecha} / \text{Total planificado del tamaño añadido y modificado a la fecha})$.</p> <p>Si solo tiene información del tamaño y del tiempo actual, coloque $\beta_0 = 0$ y $\beta_1 = (\text{Total del Tiempo de desarrollo actual a la fecha} / \text{Total actual del tamaño añadido y modificado a la fecha})$.</p>
5D	Procedimiento 5D de Estimación del tiempo	<p>Si no se tiene información histórica, use su juicio para estimar el tiempo de desarrollo a partir de tamaño estimado añadido y modificado.</p>
6	Intervalos de	<p>Si se usó el método de regresión A o B, calcule el 70% de</p>

	predicción del tiempo y tamaño	los intervalos de predicción para la estimación del tiempo y tamaño Si no se usó el método de regresión o no sabe como calcular el intervalo de predicción, calcule el mínimo y máximo y la productividad mínima para los programas escritos a la fecha.
Criterios de salida		Entradas estimadas y actuales para cada todas las categorías de tamaño pertinentes Hoja de cálculo PROBE completada con las entradas de tiempo y tamaño Valores actuales y planeados ingresados en el Resumen de plan de proyecto.

Plantilla de reporte de pruebas

Desarrollador _____ Fecha _____
 Incidencia _____ Programa# _____

Numero Prueba	_____
Objetivo	_____

Descripción	_____

Condiciones	_____

Resultados esperados	_____

Resultados actuales	_____

Numero Prueba	_____
Objetivo	_____

Descripción	_____

Condiciones	_____

Resultados esperados	_____

Resultados actuales	

Instrucciones de la plantilla de reporte de pruebas

Propósito	Mantener un registro de las pruebas realizadas y los resultados obtenidos Tener la información suficiente para que posteriormente se puedan correr nuevamente las pruebas y obtener los mismos resultados Facilitar la regresión de pruebas de un programa modificado o reusado.
General	Expandir esta tabla para múltiples copias requeridas Reportar todas las pruebas que se corrieron Sea lo más preciso y breve como sea posible
Cabecera	Introduzca su nombre y fecha. Introduzca el ID de la incidencia y numero de programa.
Nombre de la prueba	Identifique de manera única cada test de cada requerimiento. Las mismas pruebas con datos diferentes los mismos datos con diferentes pruebas
Objetivo	Brevemente describa el objetivo de la prueba
Descripción de la prueba	Describa los datos de la prueba y procedimientos en el detalle suficiente para poder realizar una regresión de pruebas posterior.
Condiciones de la prueba	Liste cualquier configuración especial, o condiciones necesarias Cuando múltiples pruebas son corridas con diferentes parámetros bajo variables condiciones, sepárelas en una lista.
Resultados esperados	Liste los resultados que se deberían producir si el test corre correctamente.
Resultados actuales	Liste los resultados que actualmente se produjeron.

Plantilla de estimación de tamaño

Desarrollador	_____	Fecha	_____
Incidencia	_____	Programa#	_____
Medida de tamaño	_____		

Partes Base	Estimado			
	Base	Eliminar	Modificar	Agregar
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
Total	B _____	D _____	M _____	BA _____

Partes Base	Actual			
	Base	Elimindadas	Modificadas	Agregadas
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
Total	_____	_____	_____	_____

Partes a agregar	Estimado				Actual	
	Tipo	Ítems	Rel. Tam	Tam.*	Tam.*	Ítems
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____	_____
Total	_____	_____	_____	PA _____	_____	_____

Partes Reusadas	Estimado Tam.	Actual Tam.
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Total	R		
Plantilla de estimación de tamaño (continuación)			
Desarrollador		Incidencia	
Hoja de cálculo PROBE (Agregado and Modificado)		Tam.	Tiempo
Tamaño agregado (A):	$A = BA + PA$		
Tamaño del proxy estimado (E):	$E = BA + PA + M$		
Método PROBE usado:	(A, B, C, o D)		
Correlación: (R^2)			
Parámetros de Regresión	β_0		
Tamaño and Tiempo			
Parámetros de Regresión:	β_1		
Tamaño and Time			
Proyectado de tamaño a agregar y modificar (P):	$P = \beta_{0_{size}}$		
+ $\beta_{1_{tamaño}} * E$			
Tamaño total estimado (T):	$T = P + B -$		
D - M + R			
Estimación total de nuevo reusable (NR):	suma de *		
ítems			
Estimación de tamaño total de desarrollo	Time =		
$\beta_{0_{tiempo}} + \beta_{1_{tiempo}} * E$			
Rango de predicción:	Rango		
Intervalo de predicción superior	UPI = P +		
Rango			
Intervalo de predicción inferior	LPI = P -		
Rango			
Porcentaje de intervalo de predicción:			

Instrucciones de la plantilla de estimación de tamaño

Propósito	Use esta plantilla con el método PROBE para realizar estimaciones de tamaño
General	<p>Una parte puede ser un modulo, componente, producto o sistema.</p> <p>Donde las partes tienen una subestructura de métodos, procedimientos, funciones o elementos similares, estos elementos de nivel más bajo se denominan ítems.</p> <p>Valores de tamaño son asumidos en la unidad especificada de tamaño</p> <p>Evitar confundir el tamaño base con el tamaño de reusó.</p> <p>Las partes a reusar no deben tener ninguna modificación</p> <p>Use el tamaño base si hay adiciones, modificaciones o eliminaciones planeadas.</p> <p>Si las partes son estimadas pero no producidas introduzca los valores actuales igual a 0.</p> <p>Si las partes son producidas pero no estimadas introduzca cero para estos valores planeados</p>
Cabecera	<p>Introduzca su nombre y fecha.</p> <p>Introduzca el ID de la incidencia y numero de programa.</p> <p>Introduzca el tipo de medida que está utilizando</p>
Partes base	<p>Si esto es la modificación o mejora de una parte existente del producto</p> <p>Mida e introduzca el valor de tamaño base (más de un producto puede ser ingresado como base)</p> <p>Estime e introduzca el tamaño de lo que va a borrar, modificar y agregar en la base del programa.</p> <p>Después del desarrollo, mida y introduzca el tamaño actual del programa base de cualquier modificación, eliminación o adición.</p>
Partes a adicionar	<p>Si usted planea agregar nuevas partes a desarrollar</p> <p>Introduzca el nombre de la parte, tipo, numero de ítems (o métodos) y tamaño relativo</p> <p>Para cada parte, obtenga el tamaño por ítem de la tabla de tamaños apropiada y multiplique este valor por el numero de ítems, y entre el tamaño estimado</p> <p>Poner un asterisco al lado del tamaño estimado de cualquier adición nueva reutilizable el tamaño actual de cada nueva parte o nuevos ítems de parte</p> <p>El numero de ítems de cada nueva parte</p>
Partes Reusadas	<p>Si usted planea incluir partes reusadas, introduzca</p> <p>Nombre de cada parte reusada sin modificaciones.</p> <p>Tamaño de cada parte no modificada.</p> <p>Después del desarrollo, entre el tamaño actual de cada parte no modificada.</p>

Instrucciones para la hoja de cálculo PROBE

Propósito	Use esta plantilla con el método PROBE para realizar la estimación de tamaño y recursos.
General	<p>El método PROBE puede ser usado por muchos tipos de estimaciones. Donde el tiempo de desarrollo esta correlacionada con el tamaño adicionado y modificado. usar la hoja de cálculo para adiciones y modificaciones introduzca los resultados de la estimación en el Resumen del plan del proyecto</p> <p>introduzca lo proyectado el valor proyectado de adiciones y modificaciones (P) en el espacio de adiciones y modificaciones el resumen del plan de proyecto</p> <p>Si el tiempo de desarrollo correlaciona con otra combinación de tipos de tamaños contables</p> <p>definir una nueva PROBE hoja de calculo</p> <p>introduzca los resultados estimados en Resumen de plan de proyecto</p> <p>use la combinación seleccionada de el tipo de tamaño contable para calcular el tamaño proyectado (P)</p> <p>introduzca el valor de P en el Resumen del plan de proyecto para el plan de tamaño apropiado de acuerdo al tipo de combinación usado</p>
Cálculos PROBE: Tamaño (Adiciones y Modificaciones)	<p>Tamaño agregado (A): Total del código base agregado (BA) y las adiciones de las partes (PA) para obtener el total de tamaño agregado (A).</p> <p>Estimar el tamaño del proxy (E): Total del tamaño adicionado (A) y modificado (M) introdúzcalo como (E)</p> <p>Método PROBE usado: Analice los datos históricos disponibles y seleccione el método apropiado PROBE (A, B, C, o D).</p> <p>Correlación: Si el método PROBE de estimación es A o B, introduzca el valor de correlación R^2) para ambos tamaño y tiempo.</p> <p>Parámetros de regresión: Siga el procedimiento PROBE para calcular los parámetros de regresión (β_0 and β_1) en introduzca los datos en los campos indicados</p> <p>Tamaño proyectado de adición y modificación (P): Usando los parámetros de regresión y el tamaño del proxy estimado, calcule el tamaño proyectado de adición y modificación (P) as</p> $P = \beta_{0Tamaño} + \beta_{1Tamaño} * E.$ <p>Estimar el tamaño total (T): Calcular el tamaño total estimado como $T = P+B-D-M+R$.</p> <p>Estimar el total nuevo reusable (NR) (NR): Totalice e introduzca los nuevos ítems reusables marcados con *</p>

Cálculos PROBE: Tiempo (Adiciones y modificaciones)	<p>Estimar basado en el método PROBE usado: Analizar los datos históricos disponibles y seleccione el método apropiado (A, B, C, o D).</p> <p>Estimar el total de tiempo de desarrollo: Usando los parámetros de regresión y el tamaño del proxy estimado (E), calcular el tiempo de desarrollo como</p> $\text{Time} = \beta_{0\text{Time}} + \beta_{1\text{Time}} * E.$
Cálculos PROBE: Rango de Predicción	<p>Calcular e introducir el rango de predicción para ambos, estimación de tamaño y tiempo.</p> <p>Calcular los intervalos superior (UPI) y e inferior (LPI) de predicción para ambos, estimación de tamaño y tiempo.</p> <p>Porcentaje Intervalo de predicción: Liste el porcentaje de probabilidad usado para calcular los intervalos de predicción (70% o 90%).</p>
Después del desarrollo (Adicionado y modificado)	<p>Introduzca los datos de tamaño actual para la base (B), eliminado (D), modificado (M), y código agregado a la base (BA), partes agregadas (PA) y partes reusadas (R).</p>

GUIONES PSP 2

SCRUM-PSP2 Guión del proceso

Propósito	Guía el desarrollo de un modulo o nivel de un programa
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Resumen del plan del proyecto SCRUM-PSP0.1 <i>Estimación por talla del requerimiento</i> <i>Datos de tamaño y tiempo históricos (Estimados y actuales)</i> <i>Logs</i> de registros de tiempos y defectos Estándares de codificación y lista de tipos de defectos Cronometro (Opcional)

Paso	Actividades	Descripción
1	Planeación	Obtenga el requerimiento del <i>sprint backlog</i> a desarrollar <i>Use el método PROBE</i> para estimar el código a adicionar y modificar en este programa <i>Complete la plantilla de estimación de tamaño</i> <i>Utilice el método PROBE</i> para estimar el tiempo requerido de desarrollo. Completar el plan de resumen del proyecto Diligenciar el tiempo en el log de registro de tiempos.
2	Desarrollo	Diseñe el programa Implemente el diseño <i>Realice la revisión de código, corrija y registre todos los defectos encontrados</i> Compile el programa, repare y registre cualquier defecto encontrado pruebe el programa, repare y registre cualquier defecto encontrado Complete el tiempo en el log de registro de tiempos.
3	Postmortem	Complete el Plan de resumen del proyecto con el tiempo actual, defectos y tamaño del programa

Criterios de Salida	Un programa probado Información del resumen de plan del proyecto completa con los datos de estimación y datos actuales <i>Plantilla de estimación de tamaño completa</i> <i>Lista de chequeo de revisión de código</i> <i>Reporte de pruebas completo</i> Registro PIP completo <i>Logs</i> de registros de tiempos y defectos completos
---------------------	---

SCRUM-PSP1 Guión de planeación

Propósito	Guiar el proceso de planeación .
Criterios de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Formulario de Resumen de plan del proyecto <i>Plantilla de estimación de tamaño</i> <i>Datos de tamaño y tiempo históricos (Estimados y actuales)</i> Log de registros de tiempo.

Paso	Actividades	Descripción
1	Estimación de tamaño	Produzca un diseño conceptual del programa Utilice el método PROBE para estimar el código a adicionar y modificar en este programa. Complete la plantilla de estimación de tamaño y el Resumen de plan del proyecto.
2	Estimación de recursos	Use el método PROBE para estimar el tiempo requerido para desarrollar este programa. Use el campo "% a la fecha" del más reciente programa desarrollado como guía para distribuir el tiempo de desarrollo en las diferentes fases.
3	<i>Estimación de defectos</i>	<i>-Basado en sus defectos a la fecha por unidad de tamaño agregado y modificado estime el total de defectos a encontrar en este programa</i> <i>-Basado en sus datos de % a la fecha, estime el numero de defectos a ser inyectados y removidos por fase</i>

Criterios de salida	Diseño conceptual del programa Plantilla de estimación de tamaño completa Resumen de plan de proyecto con los tiempos de desarrollo distribuidos, tamaño y <i>defectos</i> Diligenciar el tiempo en el log de registros de tiempos.
---------------------	---

SCRUM-PSP2.0 Guión de Desarrollo

Propósito	Guía para el desarrollo de programas pequeños.
Criterio de entrada	Requerimiento del <i>sprint backlog</i> a desarrollar Resumen de plan de proyecto con la información de tiempos. <i>Logs</i> de registro de tiempos y defectos Lista de tipos de defectos y estándares de codificación. Plantilla de diseño

Paso	Actividades	Descripción
1	Diseño	Revise los requerimientos y produzca un diseño que le permita llegar a él siguiendo las guías establecidas. Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registro de tiempos.
2	Codificación	Implemente el diseño siguiendo los estándares de codificación. Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registros de tiempos
3	Revisiones de Código	<i>Siga el guión de revisiones de código y listas de chequeo para revisar el código.</i> <i>Corrija los defectos encontrados</i> <i>Registre los defectos en el log de registro de defectos</i> <i>Registre el tiempo en el log de registro de tiempos</i>
3	Pruebas	Pruebe hasta que todos las pruebas no tengan ningún error. Repare todos los defectos encontrados Registre en el log de registro de defectos, cualquier defecto encontrado. Registre el tiempo en el log de registros de tiempos Completar la plantilla de reporte de pruebas con los test ejecutados y los resultados obtenidos.

Criterios de salida	A la salida se tendrá un programa probado y conforme a los estándares de codificación. <i>Complete la lista de chequeo de la revisión de código</i> Plantilla de reporte de pruebas completa. <i>Logs</i> de tiempos y defectos completos.
---------------------	--

PSP2 Guión Postmortem

Propósito	Guía para el proceso de Postmortem de PSP
Criterios de entrada	<p>Requerimiento del <i>sprint backlog</i> desarrollado</p> <p>Resumen del plan de proyecto con la información de tiempos</p> <p>Plantilla de reportes de pruebas completa.</p> <p><i>Logs</i> de tiempos y defectos completos</p> <p>Listas de chequeo de revisión de código completas</p> <p>Un programa probado y corriendo conforme a los estándares de codificación,</p>

Paso	Actividades	Descripción
1	Registro de defectos	<p>Revisar el Resumen del plan del proyecto para verificar que todos los defectos se encuentren registrados en todas las fases.</p> <p>Usando su mejor estimación, registre cualquier defecto omitido.</p>
2	Revisión de inconsistencia de datos en defectos	<p>Chequee que los datos de cada defecto en el log de registro de defectos se encuentre exacto y completo</p> <p>Verifique que el numero de defectos inyectados y removidos por fase sean razonables y correctos</p> <p>Determine el rendimiento del proceso y verifique que el valor es razonable y correcto</p> <p>Usando su mejor estimación, corrija cualquier pérdida o datos de defectos incorrectos.</p>
3	Tamaño	<p>Cuenta el tamaño del programa (LOC).</p> <p>Determine el tamaño del código base, borrado, modificado, adiciones a la base, reusado, nuevo reusable, y partes agregadas.</p> <p>Ingrese los datos en la plantilla de estimación de tiempo</p> <p>Determine el total de tamaño del programa</p> <p>Ingrese los datos en el Resumen de plan del proyecto.</p>
4	Tiempo	<p>Revisar en el log de registro de tiempos errores o omisiones.</p> <p>Use su mejor estimación, y corrija cualquier dato de tiempo perdido o incompleto.</p>

Criterios de Salida	<p>Un programa probado a fondo de acuerdo a los estandares de codificación.</p> <p>Listas de chequeo de revisión de código completas</p> <p>Plantilla de reporte de pruebas completa</p> <p>Completar el Resumen del plan del proyecto</p> <p>Completar el PSP describiendo los problemas, sugerencias de mejora, y lecciones aprendidas</p> <p><i>Logs</i> de registros de tiempos y defectos completos.</p>
---------------------	--

Guión de revisión de código

Propósito	Guía para la revisión de programas
Criterios de entrada	<p>Lista de chequeo de código</p> <p>Estándar de codificación</p>

		Estándar de tipos de defectos <i>Logs de registro de tiempos y defectos</i>
General		Haga la revisión de código con el código fuente
Paso	Actividades	Descripción
1	Revisión	Siga la lista de chequeo de revisión de código Revise el programa entero por cada categoría de la lista de chequeo, no intente hacer la revisión por más de una categoría al tiempo. Chequee cada ítem hasta que la lista este completa Para múltiples programas o procedimientos, utilice una lista de chequeo por separado.
2	Corrija todos los defectos	Corrija todos los defectos Si la corrección no puede completarse, aborte la revisión y retorne a la anterior fase del proceso Para facilitar el análisis de defectos, registre todos los datos especificados en el registro de defectos según las instrucciones para cada defecto.
3	Chequeo	Chequee que cada defecto fue corregido correctamente Revise nuevamente todos los cambios al diseño Registre y corrija los nuevos defectos, y donde usted sabe que el número de defectos es incorrecto corríjalo.
Criterio de salida		Una revisión completa de los fuentes del programa Una o más listas de chequeo para cada programa revisado Todos los defectos identificados y corregidos Registro de tiempo y defectos completo.

PSP2 Resumen de plan de proyecto

Desarrollador _____ Fecha _____
 Incidencia _____ Programa# _____

Resumen	Plan	Actual	To Date
LOC/Hora			
% Reusó	_____	_____	_____
% Nuevo Reusable	_____	_____	_____
Defectos en test/KLOC o equivalente	_____	_____	_____
Total defectos/KLOC o equivalente	_____	_____	_____
Rendimiento %	_____	_____	_____
Tamaño del programa	Plan	Actual	To Date
Base (B)	(Medida)	(Medida)	
Borradas (D)	(Estimada)	(Contadas)	
Modificadas (M)	(Estimada)	(Contadas)	
Agregadas (A)	(A+M - M)	(T - B + D - R)	
Reusadas (R)	(Estimada)	(Contadas)	
Agregadas y Modificadas (A+M)	(Proyectada)	(A + M)	
Tamaño Total (T)	(A+M + B - M - D + R)	(Medidas)	
Total nuevas reusables	_____	_____	_____
Tamaño del proxy estimado (E)	_____	_____	_____

Tiempos en Fase (min.)	Plan	Actual	To Date	To Date %
Planeación	_____	_____	_____	_____
Diseño	_____	_____	_____	_____
Codificación	_____	_____	_____	_____
Revisión de código	_____	_____	_____	_____
Compilación	_____	_____	_____	_____
Pruebas	_____	_____	_____	_____
Postmortem	_____	_____	_____	_____
Total	_____	_____	_____	_____

(continuación)

PSP2 Resumen de plan de proyecto (continuación)

Tiempos en Fase (min.)	<i>Plan</i>	<i>Actual</i>	<i>A la Fecha</i>	<i>A la Fecha %</i>
Planeación				
Diseño				
Codificación				
Revisión de código				
Compilación				
Pruebas				
Total Desarrollo				
Tiempos en Fase (min.)	<i>Plan</i>	<i>Actual</i>	<i>A la Fecha</i>	<i>A la Fecha %</i>
Planeación				
Diseño				
Codificación				
Revisión de código				
Compilación				
Pruebas				
Total Desarrollo				
Después del desarrollo				
<i>Eficiencia en remover defectos</i>	<i>Plan</i>	<i>Actual</i>	<i>A la fecha</i>	
<i>Defectos/Hora – Revisión de código</i>				
<i>Defectos/Hora – Compilación</i>				
<i>Defectos/Hora –Pruebas</i>				
<i>DRL (DLDR/UT)</i>				
<i>DRL (Revisión de código/UT)</i>				
<i>DRL (Compilación/UT)</i>				

Resumen del plan de proyecto PSP2

Propósito	Donde se depositara la información del plan y los datos actuales de los programas o partes de el
General	<p>Usar la más apropiada medida de tamaño, como LOC o contador de elementos.</p> <p>A la fecha es el total de valores actuales que se lleva a la fecha de todos los productos desarrollados.</p> <p><i>Una parte puede ser un modulo, componente, producto o sistema.</i></p>
Cabecera	<p>Introduzca su nombre y fecha.</p> <p>Introduzca el ID de la incidencia y numero de programa.</p>
Resumen	<p><i>Ingrese las líneas a adicionar y modificar por hora planeada, actual y a la fecha.</i></p> <p><i>% Reusado es el porcentaje de tamaño reusado de la totalidad del programa</i></p> <p>% Nuevo reusable, es el porcentaje de lo nuevo reusable agregado y modificado</p> <p>Introduzca el total de defectos estimados por línea de código</p>
Tamaño del programa	<p>Introduzca lo planeado de código base, borrado, modificado, reusado, nuevo reusable, y tamaño total desde la plantilla de estimación de tamaño</p> <p>Ingrese el plan el valor de líneas a adicionar y modificar (A+M) de lo proyectado de adicionar y modificar (P) de la plantilla de estimación de tamaño.</p> <p>Calcular el plan de tamaño a adicionar $A+M-M$.</p> <p>Ingrese el tamaño del proxy estimado (E) desde la plantilla de estimación.</p> <p>Ingrese el valor de tamaño actual del código base, borrado, modificado reusado, total y nuevo reusable. Calcule el tamaño actual agregado como $T-B+D-R$ y el actual agregado y modificado como $A+M$</p> <p>Introduzca A la fecha el tamaño de lo reusado, agregado, modificado, total y nuevo reusable.</p>
Tiempos en la fase	<p>Introduzca <i>el total de tiempo en cada fase</i> desde el total estimado de tiempo de desarrollo <i>en la plantilla de estimación de tiempo.</i></p> <p>Distribuya el estimado de tiempo total entre las fases de desarrollo de acuerdo al porcentaje a la fecha del programa desarrollado más recientemente.</p> <p>Introduzca el tiempo actual por fase y el tiempo total.</p> <p>A la fecha: Introduzca la suma de los tiempos actuales para este programa más los tiempos a la fecha del programa más reciente desarrollado.</p> <p>% A la fecha: Introduzca el porcentaje de cada fase a la fecha.</p>

PSP2 Instrucciones de plan de proyecto (continuación)

Defectos Inyectados	<p><i>Introduzca el total de defectos estimados a inyectar</i> <i>Distribuya el total de defectos estimados entre las fases acorde a los datos a la fecha del último programa desarrollado</i> Introduzca el numero de defectos actuales por fase y el total de defectos actuales. A la fecha: Introduzca la suma de los defectos actuales inyectados por cada fase y los valores a la fecha de el programa más reciente hecho previamente.. % A la fecha: Introduzca el porcentaje a la fecha de defectos inyectados en cada fase.</p>
Defectos Removidos	<p><i>Introduzca el total de defectos a remover</i> <i>Distribuya el estimado de defectos a remover entre las fases, de acuerdo a los datos históricos a la fecha del último programa desarrollado</i> A la fecha: Introduzca los defectos actuales removidos por fase más los valores a la fecha de el programa más reciente hecho previamente. % A la fecha: Introduzca el porcentaje de los defectos a la fecha removidos por fase. Después de desarrollo: Registre cualquier defecto encontrado posteriormente durante pruebas, uso, reusó o modificación.</p>
Eficiencia de remoción de defectos	<p><i>Calcule e introduzca los defectos removidos por hora en la revisión de código, compilación y pruebas</i></p>

Lista de chequeo de revisión de código

Desarrollador _____

Fecha _____

Incidencia _____

Programa# _____

Propósito	Guiar para efectuar una revisión de código efectiva
General	<p>Revise el programa entero por cada categoría de la lista de chequeo, no intente revisar más de una categoría al tiempo</p> <p>Cuando complete cada paso de la revisión, chequee el ítem en la casilla de la derecha</p> <p>Complete la lista de chequeo por cada uno de los programas o unidades antes de seguir con la siguiente.</p>

Categoría	Descripción	SI	NO
Completo	Verifique que el código cubre todo el diseño		
Inicialización	<p>Chequee la inicialización de las variables y los parámetros</p> <p>Al iniciar el programa</p> <p>Al iniciar cada ciclo</p> <p>En la entrada de cada Clase/Función /Procedimiento</p>		
Llamados	<p>Chequeo los formatos de las llamadas a las funciones.</p> <p>Parámetros del tipo correcto</p> <p>Parámetros completos</p> <p>Parámetros por referencia o valor</p>		
Nombres	<p>Chequee la ortografía de los nombres y uso</p> <p>Son consistentes?</p> <p>Su alcance es el correcto?</p>		
Objetos	<p>Chequee que</p> <p>Son inicializados correctamente</p> <p>Son eliminados solo después de su uso</p>		
Formatos de salida	<p>Chequee que el formato.</p> <p>¿Es adecuado el formato de salida de los valores?</p>		
Operadores lógicos	Verifique que se usa correctamente el operador And, Or y =		
Estándares	Garantice que el código esta conforme a los estándares de codificación		
Apertura y cierre de archivos	<p>Verifique que todos los archivos son</p> <p>Declarados apropiadamente</p> <p>Abiertos</p> <p>Cerrados</p>		

Anexo 5. Resultado respuestas encuesta equipo de desarrollo proceso de SCRUM-P

1. Considera que existe una mejor integración de SCRUM y PSP con respecto al proceso manejado anterior?

Opciones de respuesta –	Respuestas –
Si	80% 4
No	20% 1
Total	5

2. Considera que las actividades definidas en el proceso de SCRUM y PSP integrado:

Opciones de respuesta –	Respuestas –
Se Complementan	80% 4
Son Redundantes	20% 1
No tienen que ver	0% 0
Total	5

3. Considera que el proceso de SCRUM y PSP integrado esta

Opciones de respuesta –	Respuestas –
Definido claramente	40% 2
Aun existen actividades independientes	60% 3
No estan integrados	0% 0
Total	5

4. Considera que la estimación guiada por PSP con respecto a la actividad del Sprint Planning en SCRUM

Opciones de respuesta –	Respuestas –
Ayuda completamete	60% 3
Ayuda de manera parcial	20% 1
No ayuda en nada	20% 1
Total	5

5. Considera que el proceso integrado de SCRUM y PSP impacta el proceso de desarrollo

Opciones de respuesta	Respuestas
Impacta positivamente	100% 5
No lo impacta	0% 0
Impacta negativamente	0% 0
Total	5

6. Considera que PSP integrado con SCRUM impacta el control de calidad del desarrollo

Opciones de respuesta	Respuestas
Impacta positivamente	100% 5
No lo impacta	0% 0
Impacta negativamente	0% 0
Total	5

7. Considera la aplicación del proceso integrado de SCRUM y PSP dentro del proceso de desarrollo, favorece al ambiente de trabajo del equipo?

Opciones de respuesta	Respuestas
Totalmente	60% 3
Puede mejorar	40% 2
De ninguna manera	0% 0
Total	5

8. Que recomendaciones puede dar para mejorar el proceso de desarrollo que actualmente se maneja en la organización?

Respuestas
Que exista un poco mas de documentación para cada tarea y del "negocio" en general
Permitir al equipo de desarrollo ser parte del diseño de las tareas
Como las personas son reacias al cambio, se necesita acabar de tomar conciencia que estos procesos son para mejorar
Que las presiones e los clientes o las tareas "Urgentes", no desplacen los procesos que se deben realizar
Una capacitación más abierta sobre el tema
Evaluar a través de las retrospectivas los resultados de los sprint y el proceso integrado

Anexo 6 Medidas recolectadas para evaluación de calidad en funcionalidad

TAREA	CARACTERISTICA	FUNCIONALIDAD COMPLETA		FUNCIONALIDAD CON ERRORES	
		SI	NO	SI	NO
ICUATRO-2484	Se permite configurar variables de tipo check list en las variables de pruebas de calidad, e ingresar una observación en los resultados	1		1	
ICUATRO-2482	Se agrega información relacionada con documentos, pruebas de calidad, auditoria de bloqueo y desbloqueo en la ventana de administración de clasificaciones	1		1	
ICUATRO-2439	Migración de funcionalidad de generación de ordenes de producción modelo corte de Telas	1		1	
ICUATRO-2414	En la importación de documentos permite definir elementos de información de lotes para la mascara de importación	1		1	
ICUATRO-2413	Se agrega campo de clasificación a la configuración de planes de calidad, ademas se agrega filtro al listado	1		1	
ICUATRO-2412	Generación de alertas automaticas en eventos de creación de documentos, tomando la configuración de la alerta de Nuevos Documentos (Ya Definida)	1		1	
ICUATRO-2410	Permite asignar una variable de una prueba de calidad a una variable maestra	1		1	
ICUATRO-2409	Se agrega impresión de producción general en la ventana de pedidos de producción	1			1
ICUATRO-2401	Continuación de modificaciones para soporte a normas NIFF (Informes)	1		1	
ICUATRO-2395	Se agrega el control de auditoria para la información relacionada con productos controlados "Certificados y Clases Controladas"		1		1
ICUATRO-2394	Mejora en ventana de edición de no conformidades, se agrega el manejo de campos de usuario	1		1	
ICUATRO-2390	Se elimina dependencia de formulas Crystal para cantidades de informe semestral de productos controlados		1		1
ICUATRO-2389	Se agrega informe de PILA que incluye detalles por empleado según las entidades relacionadas	1		1	
ICUATRO-2385	Permite el manejo de varias cuentas de anticipos a nivel de cuentas por cobrar y cuentas por pagar	1		1	
ICUATRO-2170	Migración de funcionalidad de importación XML	1		1	
ICUATRO-2036	Ventana de gestión de certificados asignados en detalle de movimientos de documentos (Clases Controladas)	1		1	
ICUATRO-1698	Ventana de visualización de estados de producción de pedidos en rutas de procesos	1		1	
EDUWIN-219	Ajustes de compatibilidad de generación de recibos con G i4	1		1	
EDUWIN-216	Permite generación Informes de oportunidades exitosas y no exitosas,	1		1	

EDUWIN-213	Permite seleccionar el tipo de documento en los filtros de la ventana de Generación de Candidatos	1		1	
TOTAL DESARROLLOS	20	18	2	17	3
PORCENTAJE CUMPLIMIENTO		90.00%	10.00%	85.00%	15.00%

Anexo 7 Medidas recolectadas para la evaluación de calidad en rendimiento

TAREA	CARACTERISTICA	Tiempo de finalización esperado (sg)	Tiempo de finalización real (sg)	% Tiempo Adicional	Cumplio (<=10%)
ICUATRO-2484	Se permite configurar variables de tipo check list en las variables de pruebas de calidad, e ingresar una observación en los resultados	0.5	0.40	0.00%	1
ICUATRO-2482	Se agrega información relacionada con documentos, pruebas de calidad, auditoria de bloqueo y desbloqueo en la ventana de administración de clasificaciones	2	2.61	30.50%	0
ICUATRO-2409	Se agrega impresión de producción general en la ventana de pedidos de producción	1	0.86	0.00%	1
ICUATRO-2395	Se agrega el control de auditoría para la información relacionada con productos controlados "Certificados y Clases Controladas"	0.5	0.47	0.00%	1
ICUATRO-2390	Se elimina dependencia de formulas Crystal para cantidades de informe semestral de productos controlados	10	7.73	0.00%	1
ICUATRO-1698	Ventana de visualización de estados de producción de pedidos en rutas de procesos	2	1.21	0.00%	1
TOTAL DESARROLLOS	6				83.33%

TAREA	CARACTERISTICA	Procesamiento esperado (%)	Procesamiento Real (%)	% Procesamiento Adicional	Cumplio
ICUATRO-2484	Se permite configurar variables de tipo check list en las variables de pruebas de calidad, e ingresar una observación en los resultados	10	6	0.00%	1
ICUATRO-2482	Se agrega información relacionada con documentos, pruebas de calidad, auditoria de bloqueo y desbloqueo en la ventana de administración de clasificaciones	30	50.83	69.44%	0
ICUATRO-2409	Se agrega impresión de producción general en la ventana de pedidos de producción	10	10.33	3.23%	0
ICUATRO-2395	Se agrega el control de auditoria para la información relacionada con productos controlados "Certificados y Clases Controladas"	5	12.67	153.33%	0
ICUATRO-2390	Se elimina dependencia de formulas Crystal para cantidades de informe semestral de productos controlados	30	25	0.00%	1
ICUATRO-1698	Ventana de visualización de estados de producción de pedidos en rutas de procesos	10	18.67	86.67%	0
TOTAL DESARROLLOS	6				33.33%

TAREA	Aumento de consumo de memoria esperado (%)	Prom Consumo Memoria Antes de ejecutar funcion (KB)	Prom Diferencia de uso de memoria al ejecutar funcion (KB)	% aumento de consumo de memoria real (KB)	Diferencia	Sobrepaso de Consumo de memoria Esperado (%)	Cumplio
ICUATRO-2484	1.00%	110724.00	349.33	0.32%	-0.68%	0.00%	1
ICUATRO-2482	10.00%	125262.67	37378.67	29.84%	19.84%	19.84%	0
ICUATRO-2409	5.00%	141100	224	0.16%	-4.84%	0.00%	1
ICUATRO-2395	5.00%	119888	18.67	0.02%	-4.98%	0.00%	1
ICUATRO-2390	15.00%	127325.33	3389.33	2.66%	-12.34%	0.00%	1
ICUATRO-1698	15.00%	124304	74	0.06%	-14.94%	0.00%	1
TOTAL DESARROLLOS							83.33%

Anexo 8. Medidas recolectadas para evaluación de calidad en usabilidad

TAREA	Tiempo de Aprendizaje Esperado (min)	Tiempo de Aprendizaje Real (min)	% Tiempo Adicional	Requiere Demostracion	Documentada	Mensajes claros
ICUATRO-2484	5	4	0.00%	1		1
ICUATRO-2482	0	0	0.00%	1		1
ICUATRO-2439	0	0	0.00%			1
ICUATRO-2414	0	0	0.00%	1	1	1
ICUATRO-2413	0	0	0.00%			1
ICUATRO-2412	0	0	0.00%			1
ICUATRO-2410	0	0	0.00%			1
ICUATRO-2409	5	6	20.00%			1
ICUATRO-2401	0	0	0.00%			1
ICUATRO-2395	0	0	0.00%			1
ICUATRO-2394	4	5	25.00%			1
ICUATRO-2390	0	0	0.00%	1		1
ICUATRO-2389	0	0	0.00%			1
ICUATRO-2385	10	7	0.00%			
ICUATRO-2170	0	0	0.00%			1
ICUATRO-2036	0	0	0.00%			1
ICUATRO-1698	10	14	40.00%	1		1
EDUWIN-219	0	0	0.00%			1
EDUWIN-216	2	3	50.00%			1

EDUWIN-213	2	2	0.00%			1
	38	41	33.75%	25.00%	5.00%	95.00%

Anexo 9 Medidas recolectadas para evaluación de la calidad en Mantenibilidad

TAREA	CARACTERISTICA	Total de funciones	Funciones con Complejidad Ciclomática > 10	% De funciones que superan complejidad ciclomática
ICUATRO-2484	Se permite configurar variables de tipo check list en las variables de pruebas de calidad, e ingresar una observación en los resultados	172	10	5.81%
ICUATRO-2439	Migración de funcionalidad de generación de ordenes de producción modelo corte de Telas	132	5	3.79%
ICUATRO-2409	Se agrega impresión de producción general en la ventana de pedidos de producción	1	1	100.00%
ICUATRO-2170	Migración de funcionalidad de importación XML	44	3	6.82%
ICUATRO-2036	Ventana de gestión de certificados asignados en detalle de movimientos de documentos (Clases Controladas)	38	1	2.63%
ICUATRO-1698	Ventana de visualización de estados de producción de pedidos en rutas de procesos	45	0	0.00%
TOTAL DESARROLLOS	5	432	20	4.63%

TAREA	CARACTERISTICA	Numero de lineas total	Numero de lineas comentadas	% De código comentado
ICUATRO-2484	Se permite configurar variables de tipo check list en las variables de pruebas de calidad, e ingresar una observación en los resultados	3577	721	20.16%
ICUATRO-2439	Migración de funcionalidad de generación de ordenes de producción modelo corte de Telas	2378	172	7.23%
ICUATRO-2409	Se agrega impresión de producción general en la ventana de pedidos de producción	423	17	4.02%
ICUATRO-2170	Migración de funcionalidad de importación XML	1257	171	13.60%
ICUATRO-2036	Ventana de gestión de certificados asignados en detalle de movimientos de documentos (Clases Controladas)	754	63	8.36%
ICUATRO-1698	Ventana de visualización de estados de producción de pedidos en rutas de procesos	937	284	30.31%
TOTAL DESARROLLOS	5	9326	1428	15.31%

TAREA	CARACTERISTICA	Total de elementos reutilizables (Clases)	Cumple >= 1 Elemento reutilizable
ICUATRO-2484	Se permite configurar variables de tipo check list en las variables de pruebas de calidad, e ingresar una observación en los resultados	2	1
ICUATRO-2439	Migración de funcionalidad de generación de ordenes de producción modelo corte de Telas	0	0
ICUATRO-2409	Se agrega impresión de producción general en la ventana de pedidos de producción	0	0
ICUATRO-2170	Migración de funcionalidad de importación XML	0	0
ICUATRO-2036	Ventana de gestión de certificados asignados en detalle de movimientos de documentos (Clases Controladas)	1	1
ICUATRO-1698	Ventana de visualización de estados de producción de pedidos en rutas de procesos	1	1
TOTAL DESARROLLOS	5	20.00%	50.00%

TAREA	CARACTERISTICA	Número de elementos con posibles errores (20>=Complejidad ciclomatica <=30)	Cumplen =0 Elementos sin riesgos
ICUATRO-2484	Se permite configurar variables de tipo check list en las variables de pruebas de calidad, e ingresar una observación en los resultados	1	0
ICUATRO-2439	Migración de funcionalidad de generación de ordenes de producción modelo corte de Telas	2	0
ICUATRO-2409	Se agrega impresión de producción general en la ventana de pedidos de producción	1	0
ICUATRO-2170	Migración de funcionalidad de importación XML	1	0
ICUATRO-2036	Ventana de gestión de certificados asignados en detalle de movimientos de documentos (Clases Controladas)	0	1
ICUATRO-1698	Ventana de visualización de estados de producción de pedidos en rutas de procesos	0	1
TOTAL DESARROLLOS	5		33.33%

Anexo 10 Descripción del proceso de desarrollo anterior

Actividad	Descripción
Revisión Items Product Backlog	Es el proceso en el cual se realiza la revisión de los items definidos en el product backlog con el propietario del producto y el Scrum master, con el fin de establecer prioridades.
Desarrollo del Sprint Planning	Este proceso se realiza con todo el equipo y es donde se procede a exponer los requerimientos del propietario del producto al equipo para ser desarrollados en el Siguiendo Sprint, se asigna en la herramienta JIRA con el tiempo estimado al desarrollador que va a realizar la tarea desglosándola en sub tareas.
Desarrollo del Sprint	Cada desarrollador toma las tareas que tiene asignadas para realizar el respectivo desarrollo registrando los tiempos de trabajo en las tareas con las interrupciones, adicional al final del día cada desarrollador ingresa el tiempo que le resta para terminar la tarea.
Desarrollo de la Retrospectiva	Una vez se finaliza el Sprint se reúne al equipo con el fin de evaluar el desempeño del equipo en el sprint, respondiendo que estuvo bien, que se desea mejorar y que se desea probar en la siguiente iteración.
La publicación de las notas de versión	Una vez se ha finalizado el Sprint se procede a elaborar el documento donde se enuncian las nuevas funcionalidades, mejoras y corrección a bugs que se liberaron en el incremento del producto.

Anexo 11. KAs y tópicos del SWEBOK no mapeados con SCRUM y PSP

No	KA	Tópico	Sub-tópico
1	Requerimientos de Software	Fundamentos	Requerimientos de producto y proceso
			Propiedades Emergentes
			Requerimientos del sistema y del software
		Proceso	Modelos del proceso
			Gestión y soporte del proceso
			Calidad y mejora del proceso
		Análisis de Requerimientos	Modelo conceptual
			Diseño arquitectónico y asignación de requerimientos
			Negociación de requerimientos
		Especificación de requerimientos	Documento de definición del sistema
			Especificación de requerimientos del sistema
		Validación de requerimientos	Validación de modelos
			Pruebas de aceptación
		Consideraciones practicas	Naturaleza iterativa del proceso de requerimientos
			Gestión del cambio
			Atributos de los requerimientos
			Trazabilidad de requerimientos
			Métricas de requerimientos
2	Diseño de Software	Fundamentos de diseño de software	El contexto del diseño de software
			El proceso del diseño de software
		Estructura y arquitectura de software	Estructuras arquitectónicas
			Patrones macro-arquitectónicos
			Familia de programas y marcos de trabajo
		Análisis y evaluación de calidad de diseño de software	Atributos de calidad
			Métricas
		Estrategias y métodos del diseño de software	Estrategias generales
			Diseño orientado a las funciones (estructural)
			Diseño orientado a objetos
			Diseño centrado en al estructura de datos
			Diseño basado en componentes
			Otros métodos
3	Construcción del software	Consideraciones practicas	Pruebas en la construcción
			Calidad en la construcción
			Integración
4	Pruebas de Software	Técnicas de pruebas	Basada en especificaciones
			Basada en código
			Basada en fallas
			Basadas en el uso
			Basada en la naturaleza de la aplicación

			Seleccionando y combinando técnicas
		Fundamentos	Relaciones entre las pruebas y otras actividades
		Métricas relacionadas con el testing	Evaluación del programa bajo pruebas
			Evaluación de las pruebas realizadas
5	Calidad de software	Gestión del proceso de calidad de software	Asegurando la calidad de software
			Verificación y validación
			Auditorias y revisiones
		Consideraciones practicas	Requerimientos de calidad de la aplicación
			Técnicas de gestión de la calidad de software
			Métricas de la calidad de software
		Fundamentos	Cultura y ética en la ingeniería de software
			Costos y valor de la calidad
			Características y modelos de la calidad
6	Gestión en la Ingeniería de software	Promulgación del proyecto de software Inicio y definición del alcance	Gestión de contratos con proveedores
			Análisis de factibilidad
			Proceso para la validación y revisión de requerimientos
		Planeación del proyecto de software	Proceso de planeación
			Determinar entregables
			Gestión de la calidad
			Gestión del plan
		Métricas en la ingeniería de software	Estableciendo un compromiso sostenible de métricas
			Planeando el proceso de medición
			Mejorando el proceso de medición
			Evaluando las mediciones
7	Mantenimiento de Software		
8	Gestión de la configuración		
9	Proceso de Ingeniería de software		
10	Herramientas y Métodos de la Ingeniería de Software		
11	Disciplinas relacionadas con la Ingeniería del Software		

Anexo 12. Tabla de herramientas utilizadas para el proceso de evaluación de calidad

Característica Evaluada	Herramienta utilizada	Observaciones
Funcionalidad	Manual /Excel	Directamente el usuario fue quien evaluó el cumplimiento de las métricas establecidas y puntuó cada elemento en una tabla de Excel, estableciendo si los requerimientos cumplen con las especificaciones dadas.
Usabilidad	Manual / Cronometro	Se realizo a través de la toma de tiempos de aprendizaje del usuario para aprender a utilizar las nuevas funcionalidades y mejoras del <i>Sprint</i> .
Mantenibilidad	McCabe	Se realizo la medición a través de la herramienta, evaluando el código fuente generado por la construcción de las funcionalidades del <i>Sprint</i> .
Rendimiento	Monitor de Procesos S.O	Se realizo a través de la ejecución de las funcionalidades y monitorización de consumo de recursos haciendo uso del monitor de procesos del sistema operativo.